

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа информационных технологий и робототехники  
Направление подготовки 09.04.02 Информационные системы и технологии  
Отделение школы (НОЦ) информационных технологий

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
Разработка библиотеки нейросетевого распознавания рукописных символов на машиночитаемых бланках

УДК 004.932.75.032.26:371.277.2

Студент

Группа	ФИО	Подпись	Дата
8ИМ6Б	Шалаева Алёна Александровна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИТ	Лепустин Алексей Владимирович			

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент отделения социально-гуманитарных наук	Старикова Екатерина Васильевна	к.ф.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент отделения контроля и диагностики	Волков Юрий Викторович	к.т.н.		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата

## ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС ВО (ФГОС 3+), критерии АИОР
<b>Общепрофессиональные компетенции</b>		
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.	Требования ФГОС 3+ (ОПК-1, ПК 8-12, ОК-4), критерий 5 АИОР (п. 1.1), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.	Требования ФГОС 3+ (ОПК-5, ПК-7, ОК-3), критерий 5 АИОР (п. 1.1, 1.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.	Требования ФГОС 3+ (ОПК-2,6, ПК-1, ОК-1), критерий 5 АИОР (п. 1.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.	Требования ФГОС 3+ (ОПК-3,4, ПК-2,3, ОК-2), критерий 5 АИОР (п. 1.6, п. 2.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
<b>Профессиональные компетенции</b>		
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения.	Требования ФГОС 3+ (ПК-1,2,3, ОПК-2, ОК-1), критерий 5 АИОР (п. 1.3), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.	Требования ФГОС 3+ (ПК-7-13, ОПК-1, ОК-4), критерий 5 АИОР (п. 1.4), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной	Требования ФГОС 3+ (ПК-4,17, ОПК-6, ОК-4,7), критерий 5 АИОР (п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI.

	инженерии, осуществлять подготовку и обучение персонала.	Запросы студентов, отечественных и зарубежных работодателей.
P8	Формировать новые конкурентоспособные идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.	Требования ФГОС 3+ (ПК-5,6,14,15,16, ОПК-1,2, ОК-4), критерий 5 АИОР (п. 1.6), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
<b>Общекультурные компетенции</b>		
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.	Требования ФГОС 3+ (ОК-4,7, ПК-8-12, ОПК-1,6), критерий 5 АИОР (п. 2.1, п. 2.3, п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.	Требования ФГОС 3+ (ОК-3, ПК-7, ОПК-4,5), критерий 5 АИОР (п. 2.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.	Требования ФГОС 3+ (ОК-1,5, ПК-1, ОПК-2), критерий 5 АИОР (п. 2.4, п. 2.5), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.	Требования ФГОС 3+ (ОК-2,6, ПК-2,3, ОПК-3), критерий 5 АИОР (п. 2.6), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа информационных технологий и робототехники  
Направление подготовки (специальность) 09.04.02 Информационные системы и технологии  
Отделение школы (НОЦ) информационных технологий

УТВЕРЖДАЮ:  
Руководитель ООП

\_\_\_\_\_  
(Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации
--------------------------

Студенту:

Группа	ФИО
8ИМ6Б	Шалаева Алёна Александровна

Тема работы:

Разработка библиотеки нейросетевого распознавания рукописных символов на машиночитаемых бланках.	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	
--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>  <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i>	<ul style="list-style-type: none"><li>База изображений рукописных символов;</li><li>Машиночитаемые бланки;</li></ul>
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i>	<ul style="list-style-type: none"><li>Анализ предметной области;</li><li>Определение оптимальных параметров для реализации сверточной нейронной сети;</li><li>Разработка библиотеки сверточной нейронной сети.</li></ul>

<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>		Презентация в формате *.pptx
<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>		
<b>Раздел</b>	<b>Консультант</b>	
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Старикова Екатерина Васильевна	
Социальная ответственность	Волков Юрий Викторович	
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>		
Обзор алгоритмов и библиотек нейросетевого распознавания (Recognition algorithms review)		

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	
---	--

**Задание выдал руководитель:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Старший преподаватель отделения ИТ	Лепустин А.В.			

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ИМ6Б	Шалаева А.А.		

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Школа информационных технологий и робототехники  
Направление подготовки (специальность) 09.04.02 Информационные системы и технологии  
Уровень образования магистр  
Отделение школы (НОЦ) информационных технологий  
Период выполнения \_\_\_\_\_ (осенний / весенний семестр 2017/2018 учебного года)

Форма представления работы:

магистерская диссертация
--------------------------

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**  
**выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
7.02.18	Анализ предметной области	10
25.02.18	Проектирование библиотеки нейросетевого распознавания	15
10.05.18	Разработка библиотеки нейросетевого распознавания	30
15.05.18	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
18.05.18	Социальная ответственность	10
21.05.18	Обязательное приложение на иностранном языке	10
24.05.18	Оформление пояснительной записки	15

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИТ	Лепустин А. В.			

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
	Шерстнёв В. С.	к.т.н, доцент		

**Форма задания для раздела магистерской диссертации**  
**«Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»**

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА**  
**«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И**  
**РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ИМ6Б	Шалаева Алёна Александровна

Школа	Информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	магистратура	Направление/специальность	09.04.02 «Информационные системы и технологии»

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого и инновационного потенциала НТИ	Оценка конкурентоспособности разработки, анализ перспективности проекта
2. Организация и планирование работ	Планирование этапов разработки программы, определение трудоемкости, построение диаграммы Ганта.
3. Формирование бюджета научных исследований	Расчёт сметы затрат на выполнение проекта
2. Определение ресурсной, финансовой, экономической эффективности	Оценка научно-технического уровня разработки

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. Сегментирование рынка
2. Оценка конкурентоспособности технических решений
3. QuaD-анализ
4. Матрица SWOT
5. График проведения и бюджет НТИ
6. Оценка ресурсной, финансовой и экономической эффективности НТИ

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Старикова Е.В.	к.ф.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ6Б	Шалаева Алёна Александровна		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ6Б	Шалаева Алёна Александровна

Школа	Информационных технологий и робототехники	Отделение	Информационных технологий
Уровень образования	магистратура	Направление/специальность	09.04.02 «Информационные системы и технологии»

### Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является разработка библиотеки нейросетевого распознавания рукописных символов на машиночитаемых бланках. Методика – разработка программного обеспечения. Рабочая зона – ТОИПКРО.
--	--

### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<b>1. Производственная безопасность</b> 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности: <ul style="list-style-type: none"> <li>– физико-химическая природа вредности, её связь с разрабатываемой темой;</li> <li>– действие фактора на организм человека;</li> <li>– приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ);</li> <li>– предлагаемые средства защиты;</li> <li>– (сначала коллективной защиты, затем – индивидуальные защитные средства).</li> </ul> 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности: <ul style="list-style-type: none"> <li>– механические опасности (источники, средства защиты);</li> <li>– термические опасности (источники, средства защиты);</li> <li>– электробезопасность (в т.ч. статическое электричество, молниезащита – источники, средства защиты);</li> <li>– пожаровзрывобезопасность (причины, профилактические мероприятия, первичные средства пожаротушения).</li> </ul>	1.1. Анализ вредных и опасных факторов, которые может создать объект исследования. 1.2. Анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте при проведении исследований. 1.3. Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов.
<b>2. Экологическая безопасность:</b> <ul style="list-style-type: none"> <li>– защита селитебной зоны</li> </ul>	2.1. Анализ возможного влияния объекта исследования на окружающую среду.



<ul style="list-style-type: none"> <li>– анализ воздействия объекта на атмосферу (выбросы);</li> <li>– анализ воздействия объекта на гидросферу (сбросы);</li> <li>– анализ воздействия объекта на литосферу (отходы);</li> <li>– разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</li> </ul>	<p>2.2. Анализ влияния процесса исследования на окружающую среду.</p> <p>2.3. Обоснование мероприятий по защите окружающей среды.</p>
<p><b>3. Безопасность в чрезвычайных ситуациях:</b></p> <ul style="list-style-type: none"> <li>– перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</li> <li>– выбор наиболее типичной ЧС;</li> <li>– разработка превентивных мер по предупреждению ЧС;</li> <li>– разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</li> </ul>	<p>3.1. Анализ вероятных ЧС, которые может инициировать объект исследований.</p> <p>3.2. Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследований.</p> <p>3.3. Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС.</p>
<p><b>4. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<p>4.1. Специальные (характерные для рабочей зоны исследователя) правовые нормы трудового законодательства. 4.2. Организационные мероприятия при компоновке рабочей зоны исследователя.</p>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент отделение контроля и диагностики	Волков Юрий Викторович	к.т.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ИМ6Б	Шалаева Алёна Александровна		

## РЕФЕРАТ

Ключевые слова: информационная система, искусственные нейронные сети, сверточные нейронные сети, алгоритм обратного распространения ошибки, обучающая выборка, формальный нейрон, функция активации.

Объект исследования является разработанная сверточная нейросеть для распознавания объектов - рукописных символов.

Цель работы: Разработка библиотеки нейросетевого распознавания рукописных символов на машиночитаемых бланках.

В процессе работы был сделан обзор архитектур нейронных сетей, анализ алгоритмов обучения и уже существующих библиотек, показывающий их сильные и слабые стороны. Были выявлены оптимальные параметры для реализации сверточной нейронной сети.

В результате исследования, выбрана наиболее подходящая нейросетевая архитектура для реализации поставленной задачи, спроектирована архитектура программы, реализованы классы и методы необходимые для обучения и функционирования сверточной сети.

Область применения: распознавание бланков ГИА и ЕГЭ.

В последующей разработке планируется усовершенствовать алгоритм распознавания, что приведет к повышению точности до 98%. Также, предполагается дальнейшая интеграция программы в стороннюю систему, при удовлетворяющем результате распознавания.

## ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ

В данной работе применены следующие термины с соответствующими определениями:

– *Машинное обучение* – обширный подраздел искусственного интеллекта, математическая дисциплина, использующая разделы математической статистики, численных методов оптимизации, теории вероятностей, дискретного анализа, и извлекающая знания из данных;

– *Глубокое обучение* – это часть более широкого семейства методов машинного обучения – обучения представлением, где векторы признаков располагаются сразу на множестве уровней. Эти признаки определяются автоматически и связывают друг с другом, формируя выходные данные;

– *Искусственная нейронная сеть* – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма;

– *Искусственный нейрон* – узел искусственной нейронной сети, являющийся упрощённой моделью естественного нейрона;

– *Синапс* – это связь между двумя нейронами. У синапсов есть один параметр -вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому;

– *Слой* – один или несколько нейронов, на входы которых подается один и тот же общий сигнал;

– *Архитектура нейронной сети* – описание слоёв сети с указанием их типа и связей между ними из которого ясна последовательность обработки исходного сигнала. Может быть описана формально на одном из языков программирования или схематически;

– *Настройки нейронной сети* – набор параметров нейронной сети не относящихся к описанию архитектуры: размер ядра свёртки, смещение ядра

свёртки, активационная функция, количество карт признаков генерируемых на слое и т. д.;

- *Эпоха* – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и, возможно, проверку качества обучения на контрольном множестве;

- *Функция ошибок* – это целевая функция, требующая минимизации в процессе управляемого обучения нейронной сети. С помощью функции ошибок можно оценить качество работы нейронной сети во время обучения. Например, часто используется сумма квадратов ошибок;

- *Полносвязность* – в полносвязных сетях(слоях) каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам;

- *Коэффициент скорости обучения* – параметр градиентных методов обучения нейронных сетей, позволяющий управлять величиной коррекции весов на каждой итерации.

В настоящей работе использованы следующие сокращения:

- *ИНС(ANN)* – искусственная нейронная сеть;
- *СНС (англ., CNN)* – сверточная нейронная сеть;
- *ПНС(англ. FCN)* – полносвязно-свёрточная сеть;
- *CPU (англ. central processing unit)* – центральный процессор;
- *SGD* – метод стохастического градиентного спуска;
- *Adam* – метод адаптивной инерции;
- *Adagrad* – метод адаптивного градиента;
- *Adadelta* – метод адаптивного шага обучения;
- *ЕГЭ* – единый государственный экзамен;
- *ГИА* – государственная итоговая аттестация;
- *ОС (англ., OS)* – операционная система.

## Содержание

ВВЕДЕНИЕ .....	15
1 ОБЗОР АЛГОРИТМОВ И БИБЛИОТЕК НЕЙРОСЕТЕВОГО РАСПОЗНАВАНИЯ.....	17
1.1 Искусственные нейронные сети.....	17
1.1.1 Модель формального нейрона .....	17
1.1.2 Функция активации.....	18
1.1.3 Алгоритм обучения нейронной сети.....	21
1.2 Архитектура нейронных сетей.....	24
1.2.1 Нейросети прямого распространения .....	24
1.2.2 Рекуррентные нейросети .....	25
1.2.3 Сверточные нейронные сети.....	26
1.3 Обзор библиотек машинного обучения .....	30
2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ.....	34
2.1 Обучающая выборка .....	34
2.2 Определение оптимальных параметров для реализации CNN .....	37
2.3 Проектирование логической модели системы .....	40
3 РЕЗУЛЬТАТЫ ПРОВЕДЕННОГО ИССЛЕДОВАНИЯ .....	43
4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	44
4.1 Предпроектный анализ .....	44
4.1.1 Потенциальные потребители результатов исследования .....	44
4.1.2 Анализ конкурентных технических решений .....	46
4.1.3 QuaD-анализ.....	47
4.1.4 SWOT-анализ.....	49
4.2 Определение возможных альтернатив разработки .....	50
4.3 Организация и планирование работ .....	51
4.3.1 Продолжительность этапов работ .....	53
4.3.2 Расчет накопления технической готовности .....	57
4.4 Бюджет научно-технического исследования.....	58
4.4.1 Расчет материальных затрат.....	58
4.4.2 Расчет основной заработной платы исполнителей системы .....	59
4.4.3 Расчет затрат по дополнительной заработной плате.....	60
4.4.4 Расчет отчислений во внебюджетные фонды .....	61
4.4.5 Расчет накладных расходов.....	61
4.4.6 Формирование бюджета затрат проекта .....	62
4.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования ..	62
4.6 Вывод .....	65
5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....	67
5.1 Производственная безопасность .....	67
5.2 Вредные производственные факторы.....	68
5.2.1 Повышенный уровень электромагнитных излучений .....	68

5.2.2 Повышенные показатели микроклимата .....	69
5.2.3 Повышенный уровень шума .....	70
5.2.4 Недостаточность освещенности рабочей зоны.....	70
5.2.5 Психофизические факторы .....	71
5.3 Опасные производственные факторы .....	72
5.3.1 Опасность поражения электрическим током .....	72
5.3.2 Опасность возникновения пожара.....	73
5.4 Экологическая безопасность .....	73
5.5 Безопасность в чрезвычайных ситуациях .....	74
5.6 Правовые и организационные вопросы обеспечения безопасности...	74
ЗАКЛЮЧЕНИЕ .....	76
ПРИЛОЖЕНИЕ А .....	80
ПРИЛОЖЕНИЕ Б.....	90

## ВВЕДЕНИЕ

В настоящее время для решения многих практических задач в различных сферах человеческой деятельности широкое распространение получили нейронные сети. Данные модели искусственного интеллекта используются при диагностике, прогнозировании, в решении оптимизационных задач, а также в задачах распознавания образов и анализа изображений. Обладая высокой аппроксимирующей способностью, нейронная сеть, обучившись на экспериментальных данных, способна решать поставленные задачи с высокой степенью точности.

Целью данной работы является разработка библиотеки нейросетевого распознавания символов на машиночитаемых бланках.

Для достижения поставленной цели необходимо решить ряд задач: изучение и анализ предметной области, выявление и документирование требований к разрабатываемой библиотеке, проектирование архитектуры ИНС, реализация функциональных возможностей и пользовательского интерфейса.

Для решения данных задач, используются тип нейронных сетей, называемый сверточным. Сверточная нейронная сеть (СНС) состоит из серии слоев. Берётся изображение, пропускается через чередование свёрточных, нелинейных слоев, и с помощью полносвязного слоя порождается вывод. В качестве вывода может выступать класс или вероятность класса, которое лучше всего описывает изображение.

В работе был проведен анализ уже существующих библиотек нейросетевого распознавания. Вследствие, было принято решение разработки данной библиотеки, позволяющей упростить процесс ее интеграции с подсистемой предобработки и сегментации.

Объектом исследования в данной работе являет разработанная сверточная нейросеть для распознавания объектов - рукописных символов.

Предметом исследования упает разработка программного продукта, содержащего в себе библиотеку нейросетевого распознавания.



# 1 ОБЗОР АЛГОРИТМОВ И БИБЛИОТЕК НЕЙРОСЕТЕВОГО РАСПОЗНАВАНИЯ

Данный обзор представляет собой набор ключевых моментов, касающиеся ИНС и метод их обучения. Например, с помощью обратного распространения ошибки [4, 5, 6]. Так же приведены виды архитектуры ИНС [1,2], каждая из которых подходит для определенного класса задач. Представлен анализ существующих библиотек нейросетевого распознавания рукописных символов.

## 1.1 Искусственные нейронные сети

В настоящий момент, ИНС активно применяется в диагностике, в решении оптимизационных задач, для распознавания образов и анализа изображений. ИНС представляет собой динамическую систему, состоящую из связанных меж собой элементарных процессов – формальных нейронов.

### 1.1.1 Модель формального нейрона

Модель формального нейрона, составляет основу структурной и функциональной части ИНС. Такой нейрон представляет элементарный вычислитель (процессор), применяемый в узлах нейронной сети.

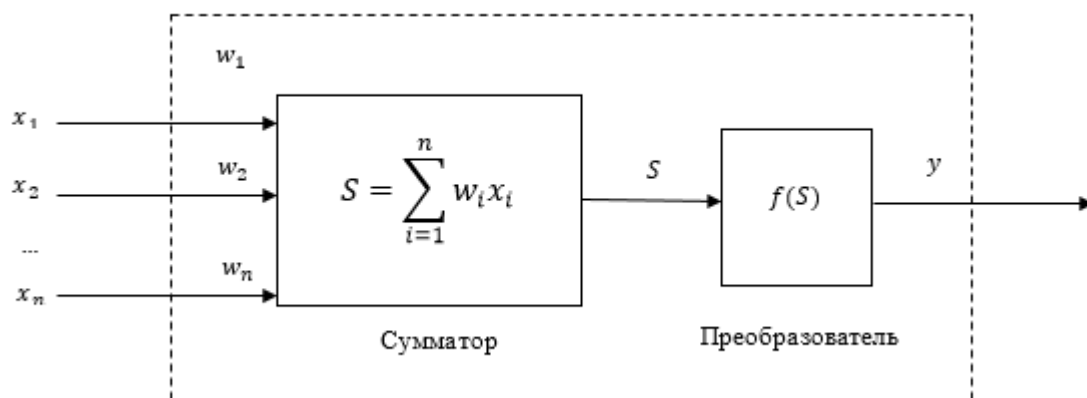


Рисунок 1 – Модель формального нейрона

Математическая модель нейрона:

$$S = \sum_{i=1}^n w_i x_i + b, \quad (1)$$

$$y = f(S), \quad (2)$$

где  $x_i$  – элемент вектора входного сигнала;

$w_i$  – значение веса связи нейрона;

$b$  – смещение нейрона;

$y$  – выходной сигнал.

В нейрон поступают входные сигналы ( $x_1 \dots x_n$ ) через линейные связи – синапсы. Каждый синапс имеет вес ( $w_1 \dots w_n$ ) связи между двумя нейронами, имитируя пропускную способность в биологических нейронах. Следующим важным элементом ИНС является адаптивный сумматор. Основной задачей последнего является суммирование входных сигналов нейронов, предварительно помноженных на соответствующие веса связей. Итогом работы адаптивного сумматора выступает взвешенная сумма, характеризующая поступивший на нейрон сигнал в целом. Эту сумму необходимо обработать нейрону для формирования выходного сигнала, с помощью функции активации (передаточной функции). Последняя принимает сумму, как аргумент. В результате, значение функции активации является выходным сигналом нейрона.

### 1.1.2 Функция активации

Функция вычисляет выходной сигнал нейрона на основе входных сигналов и взвешенной суммы соответственно.

Перечень основных видов [1, 4]:

**1. Пороговая функция (функция Хевисайда).** Описывается:

$$f(S) = \begin{cases} 0, & \text{при } S < 0 \\ 1, & \text{при } S \geq 0 \end{cases} \quad (1)$$

Не предназначена для алгоритма обратного распространения ошибки.

**2. Линейная функция.** Описывается:

$$f(S) = S \quad (2)$$

**3. Кусочно-линейная функция.** Описывается:

$$f(S) = \begin{cases} 1, & S \geq \frac{1}{2} \\ |S|, & \frac{1}{2} > S > -\frac{1}{2} \\ 0, & S \leq -\frac{1}{2} \end{cases} \quad (3)$$

**4. Сигмоидальная логистическая функция.**

$$f(S) = \frac{1}{1 + e^{(-aS)}}, \quad (4)$$

где,  $a$  – параметр наклона, изменение которого позволит построить функцию с различной крутизной наклона. Область значений функции лежит в диапазоне от 0 до 1.

**5. Выпрямленная линейная функция активации (ReLU).** Является простой и эффективной с точки зрения вычислительной сложности вариантом передаточной функции. На рисунке 2 представлен график функции ReLU и ее первой производной. Производная может быть равна двум значениям единице или нулю, в следствие чего не происходит разрастания или затухания градиентов.

$$f(S) = \max(0, S) = \begin{cases} 0, & \text{при } S < 0 \\ S, & \text{при } S \geq 0 \end{cases} \quad (5)$$

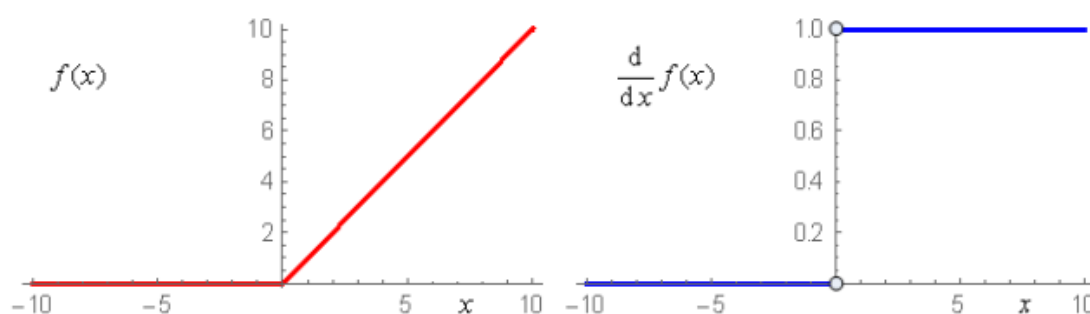


Рисунок 2 – График функции ReLU

*Достоинства функции ReLU:*

- Не имеет ресурсоемких операций;
- Убирает не нужные детали;
- Отсутствует разрастание и затухание градиента;
- Быстро обучается.

*Недостатки:*

- Не всегда надежна в процессе обучения;
- Сильная зависимость от инициализации весов.

**6. Функция Softmax.** Разработана, с целью превращения вектора с реальными значениями в вектор вероятностей и определяется для  $i$ -ого нейрона как:

$$z_i = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)}, \quad (6)$$

Где  $z_i$  – искомое значение выхода  $i$ -ого нейрона;

$y_i$  – исходное значение выходного сигнала.

Преимущество данной функции основывается на частной производной  $i$ -ого нейрона:

$$\frac{\partial z_i}{\partial y_i} = z_i(1 - z_i), \quad (7)$$

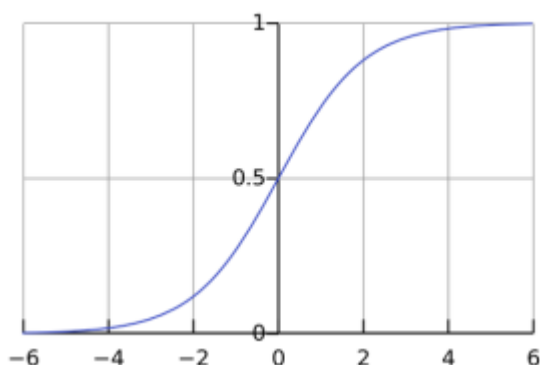


Рисунок 3 – График функции Softmax

### 1.1.3 Алгоритм обучения нейронной сети

Самым распространённым алгоритмом обучения нейросетей – метод обратного распространения ошибки (метод градиентного спуска). Основная идея: при известной ошибке и производной сжимаемой функции, решение задачи реализуется в обратную сторону, при этом рассчитать насколько необходимо изменить веса для минимизации ошибки. Данный алгоритм относится к методам обучения с учителем [4, 5, 6]. Есть набор входных данных  $X$  и ответы на данные  $Y$ , необходимо правильно подобрать параметры модели, для наилучшего описания исследуемых данных. Чтобы настроить сеть для поставленной задачи нужно: минимизировать отклонение спрогнозированных ответов на полученных данных реальных ответов, при помощи среднеквадратичной функции эмпирического риска:

$$\hat{C}(f) = \frac{1}{N} \sum_{i=1}^N (f(x_i, \theta) - y_i)^2, \quad (8)$$

где  $\theta$  - параметр, по которому осуществляется оптимизация, при зафиксированной  $f$ .

Процесс обучения ИНС реализуется следующим образом: в качестве входных сигналов выступают примеры из обучающей выборки. Одна такая итерация называется эпохой. Обучение продолжается от эпохи к эпохе до тех пор, пока веса нейронов и уровни порога ИНС не урегулируются, и ошибка сети не достигнет заданного минимума.

Реализация выполняется одним из следующих способов:

- Стохастический градиентный спуск (последовательный режим). Суть данного способа состоит в том, что после прохода каждого примера обучающей выборки по НС, веса нейронов корректируются.
- Пакетный режим. Изменение весов связей происходит только после подачи на вход всей обучающей выборки одной эпохи.

– Mini-batch. Представляет собой компромиссное решение между последовательным и пакетным режимом. Корректировка весов происходит после подачи на вход части обучающей выборки одной эпохи.

Далее детальнее рассматривается метод SGD и его наиболее распространенных модификаций.

*Стохастический градиентный спуск (SGD)* [8] каждый параметр обновляется, благодаря вычитанию градиента оптимизируемой функции по соответствующему параметру и масштабируя его на шаг обучения  $\eta$ , являющимся гиперпараметром. При большом значении  $\eta$ , метод расходится. При слишком маленьком значении – сходится медленно. Правило пересчета:

$$i \sim u \{1, 2, \dots, n\}$$

$$\theta_{t+1} = \theta_t - \eta \nabla f_i(\theta_t), \quad (9)$$

где  $\theta$  - параметр, по которому осуществляется оптимизация, при зафиксированной  $f_i$ ;

$f_i$  – функция, подсчитанная на основе  $i$ -ой мини-части данных;

$i$  – индекс выбранный случайным образом;

$\eta$  - шаг обучения.

*Метод адаптивного градиента (AdaGrad)* [9] изменяет шаг обучения для каждого параметра в отдельности, учитывая историю всех прошлых градиентов для этого параметра. Реализуется делением каждого элемента в градиенте  $\nabla f_i$  на квадратный корень суммы квадратов прошлых соответствующих элементов градиента. Таким образом уменьшается шаг обучения для параметров, имеющих большую величину градиента. Формулы для пересчета:

$$g_{t+1} = g_t + \nabla f_i(\theta_t)^2 \quad (10)$$

$$\theta_{t+1} = \theta_t - \frac{\eta \nabla f_i(\theta_t)}{\sqrt{g_{t+1} + \epsilon}}, \quad (11)$$

где  $\epsilon$  — константа, выделенная для численной стабильности.

$\eta$ - шаг обучения;

$\theta_t$  — параметр оптимизации;

$f_i$  — функция, подсчитанная на основе  $i$ -ой мини-части данных;

*Метод адаптивного шага обучения (AdaDelta)* [10] использует экспоненциальное скользящее среднее для оценки второго момента градиента  $g_t$ . Так же метод вычисляет скользящее среднее  $x_i$  и  $v_i$  аналогичным инерции, но при обновлении этой величины, используется квадрат текущего шага. Обновление параметров происходит по формулам 11 и 12:

$$g_{t+1} = \gamma g_t + (1 - \gamma) \nabla f_i(\theta_t)^2, \quad (12)$$

где  $g_t$  - второй момент градиента;

$f_i$  — функция, подсчитанная на основе  $i$ -ой мини-части данных;

$\theta_t$  — параметр оптимизации.

*Метод адаптивной инерции (Adam)* [11] метод аналогичен двум методам, описанным выше, но отличается следующим: оценкой первого момента вычисляется как скользящее среднее, и из-за того, что оценки первого и второго моментов инициализируются нулями, используется небольшая коррекция, чтобы результирующие оценки не были смещены к нулю. При заданных параметрах  $\gamma_1, \gamma_2, \eta, \lambda, m_0 = 0, g_0 = 0$ , правило пересчета:

$$m_{t+1} = \gamma_1 m_t + (1 - \gamma_1) \nabla f_i(\theta_t) \quad (13)$$

$$g_{t+1} = \gamma_2 g_t + (1 - \gamma_2) \nabla f_i(\theta_t)^2 \quad (14)$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \gamma_1^{t+1}} \quad (15)$$

$$\hat{g}_{t+1} = \frac{m_{t+1}}{1 - \gamma_2^{t+1}} \quad (16)$$

$$\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{g}_{t+1} + \epsilon}} \quad (17)$$

## 1.2 Архитектура нейронных сетей

Существуют различные архитектуры ИНС. Несмотря на их различие они обладают некоторыми общими свойствами:

- ИНС состоит из формальных нейронов;
- Обработка сигнала производится параллельно, за счет слоев, состоящих из нейронов.

Сложность ИНС измеряется количеством слоев и нейронов. Чем сложнее ИНС, тем обширнее область ее применения. Но нужно учитывать ограниченность ресурсов ПЭВМ.

### 1.2.1 Нейросети прямого распространения

Классической архитектурой считаются полносвязные НС прямого распространения. Представляет собой многослойный персептрон – многослойная полносвязная сеть, на рисунке 4. Такая архитектура подходит для работы с классификацией, однако имеет следующие проблемы:

- Множество параметров. Сеть имеющая множество параметров, дополнительно переобучается. В результате может обучиться тому, чего нет. Например, шум DataSet.
- Затухающие градиенты. Когда НС многослойны, то от градиента в самом конце может остаться только малая часть. В следствии чего, изменение весов на входе практически невозможно.



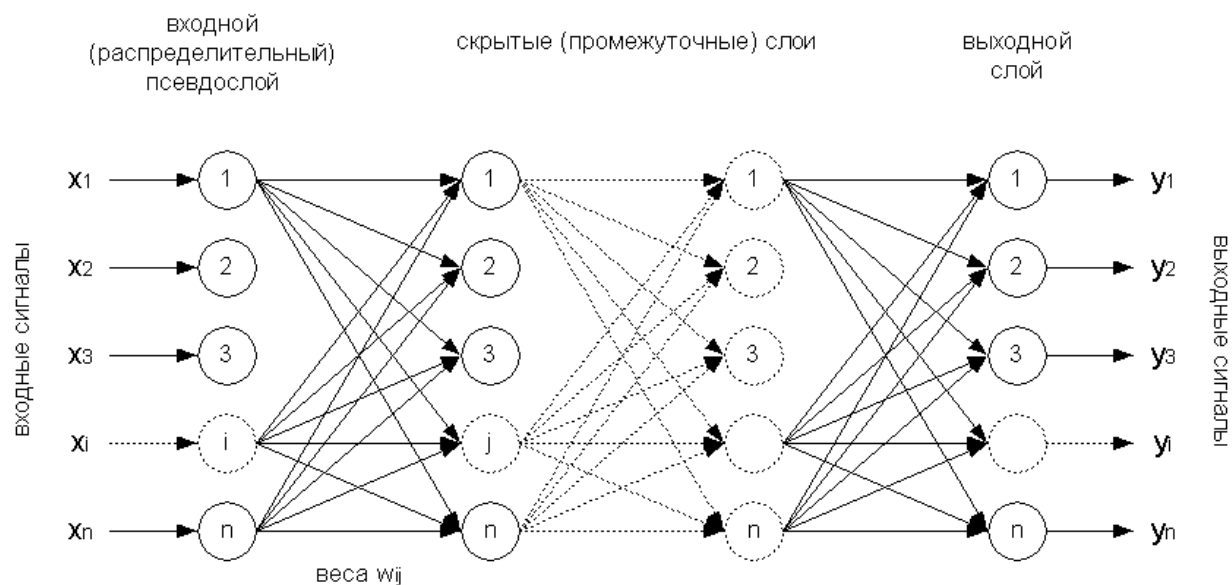


Рисунок 4 – Многослойный персептрон

### 1.2.2 Рекуррентные нейросети

Рекуррентная нейронная сеть отличается от обычных сетей наличием циклической связи. Скрытый слой, на следующем шаге, отправляет собственные значения к самому себе. РНС позволяет реализовать любую вычислительную функцию, обладают определенным видом памяти и лучше подходят для работы с последовательностями, моделированием контекста и временными зависимостями. Схематическое отображение данной сети представлено на рисунке 5.

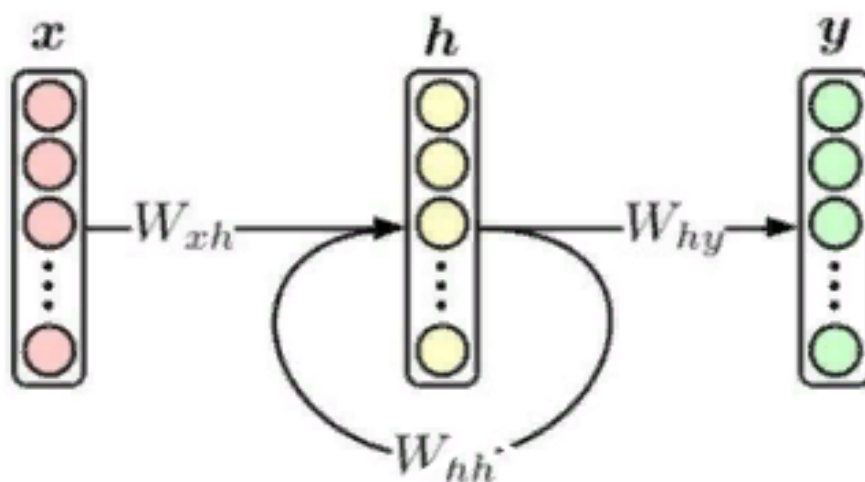


Рисунок 5 – Рекуррентная нейронная сеть

### 1.2.3 Сверточные нейронные сети

Сверточные нейронные сети представляют собой ИНС направленные на распознавание объектов на изображении, так же она является главной составляющей технологии глубокого обучения. Главным преимуществом СНС стало концепция общих весов [3]. Сети можно настраивать небольшим количеством параметров, несмотря на их размерность. СНС устойчивы к изменениям масштаба, смещениям, поворотам, смене ракурса и другим искажениям. Схематическое отображение данной сети представлено на рисунке 6.

СНС состоит из следующих слоёв:

- 1) Входной слой;
- 2) Сверточные слои (convolutional);
- 3) Pooling-слои (Subsampling);
- 4) Полносвязный слой;
- 5) Выходной слой.

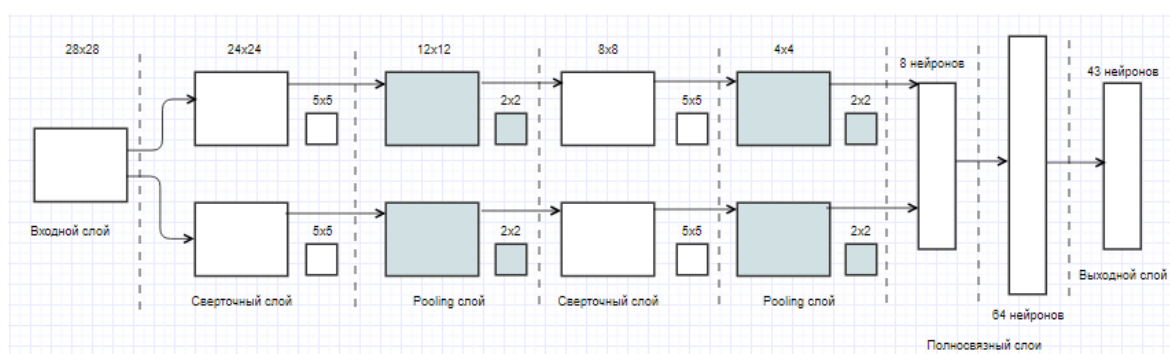


Рисунок 6 – Структура сверточной нейросети

1) Входной слой. Данные, поступающие на вход, представляют собой изображения типа bmp, размера 28x8 пикселей. При большой размерности, то вычислительная сложность повышается, что влияет на ограничение в скорости ответа. Определение размера решается методом подбора.

2) Сверточный слой (convolutional). Сверточный слой представляет из себя набор карт (другое название – карты признаков, в обиходе это обычные матрицы), у каждой карты есть синаптическое ядро (матрицы, именуемая ядром свертки). Сверточный слой можно назвать матричным фильтром, трансформирующим исходное изображение. При реализации сверточного слоя, происходит наложение ядра свертки на фрагмент изображения, соответствующие элементы перемножаются. Полученные значения суммируются и фиксируются в центральной ячейке матрицы. Целью обучения сверточного слоя является нахождение ядра свертки, полностью эквивалентного всем весам нейронов. Данную операцию можно реализовать несколько раз. Визуализация данного слоя представлена на рисунке 7.

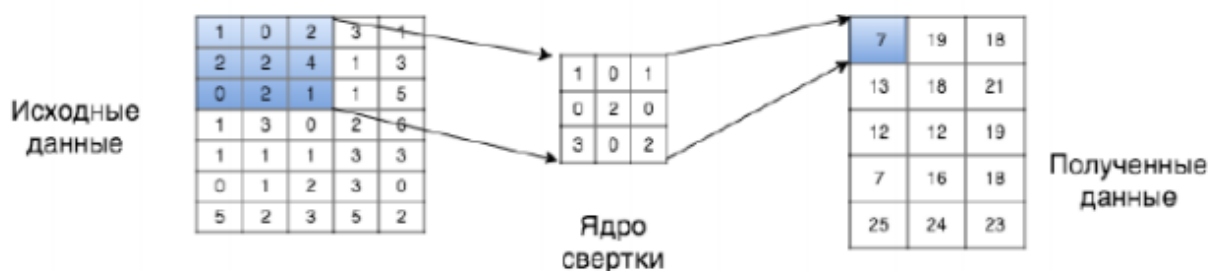


Рисунок 7 – Визуализация операции свёртки

3) Pooling-слой. Подвыборочный слой уменьшает размерность входной карты признаков. Уменьшение размерности реализуется различными способами. Самый знаменитый – метод выбора максимального элемента, расположенного около текущего элемента. Такие типы слоев convolutional и pooling, чередуясь между собой, формируют входной вектор признаков для многослойного персептрона.

Подвыборочный слой позволяет:

- увеличить скорость вычислений (минимум в 2 раза), за счет уменьшения размерности карт предыдущего слоя;
- осуществлять фильтрацию уже ненужных компонентов;
- осуществлять поиск признаков более высокого уровня, используемых для следующего сверточного слоя.

Подвыборочный слой содержит карты, как и сверточный, количество которого совпадает с предыдущим (сверточным) слоем, их 6. Данный слой уменьшает размер карт предыдущего слоя. При выявлении некоторых признаков уже выявленных на предыдущем, для дальнейшей обработки детальное изображение не используется, и оно сжимается до менее детального. Фильтрация не нужных деталей позволяет сети не переобучаться.

При сканировании матрицей свертки подвыборочного слоя карты предыдущего слоя, сканирующая матрица не пересекается, как в сверточном слое. Стандартно, карта имеет матрицу свертки размером  $2 \times 2$ , что уменьшает предыдущие карты сверточного слоя в 2 раза. Карта признаков подразделяется на ячейки  $2 \times 2$  компонент, из которых выбираются максимальные. Визуализация данного слоя представлена на рисунке 8.



Рисунок 8 – Визуализация работы слоя Pooling

4) Полносвязный слой. Заключается в классификации, имитируя сложную нелинейную функцию, реализация которой приводит к повышению точности распознавания. Представляет собой многослойный персептрон, накладываемый на первые два слоя. Полносвязный слой реализуется соединением нейронов текущего слоя с нейронами на предшествующем. Пример полносвязного слоя приведен на рисунке 9.

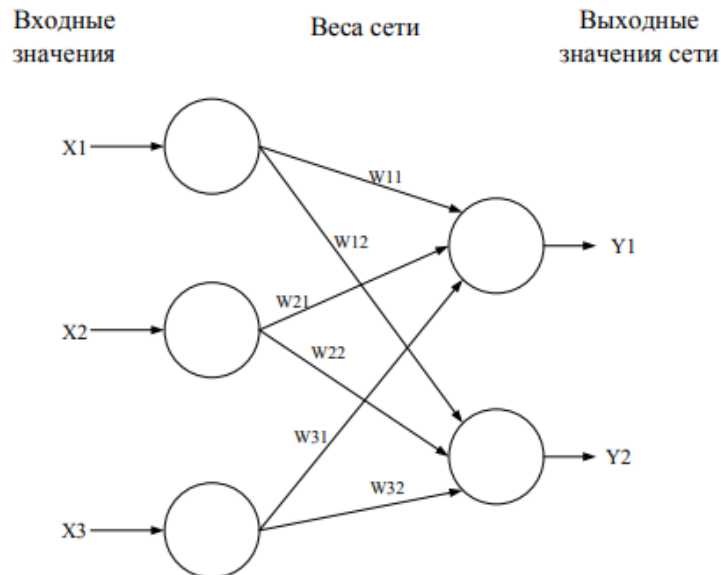


Рисунок 9 – Структура полносвязного слоя

5) Выходной слой. Выходной слой связан со всеми нейронами предыдущего слоя. Количество нейронов соответствует количеству распознаваемых классов.

## Преимущества и недостатки СНС

### Преимущества [5]:

- справляется с задачей классификации и распознаванию изображений;
- число настраиваемых параметров, по сравнению с полносвязной нейронной сетью меньше;
- высокая устойчивость к сдвигам, поворотам поступающего на вход изображения, по сравнению с ПНС.

**К недостаткам** относятся следующие настраиваемые параметры:

- число сверточных и подвыборочных слоев;
- количество карт признаков;
- размерность ядра;
- функция активации;
- скорость обучения СНС.

Вышеперечисленные параметры сильно влияют на результат работы СНС, чаще всего выбираются эмпирически. Для популярных задач существуют определенные конфигурации сетей, но для новых задач подбор гиперпараметров осуществляется опытным путем.

### 1.3 Обзор библиотек машинного обучения

Существует множество программных средств для решения задач машинного обучения, в таблице 1 приведены наиболее популярные.

Таблица 1 – Характеристики существующих библиотек

№	Название	Язык	ОС	FC NN	CNN	AE	RBM
1	Theano	Python	Windows, Linux, Mac	+	+	+	+
2	Pylearn2	Python	Windows, Linux, Vagrant	+	+	+	+
3	Torch	Lua, C	Linux, Mac, Android	+	+	+	+
4	Caffe	Python, C++	Linux	+	+	-	-

В качестве исходных данных будет использоваться база изображений рукописных символов, предоставленная предприятием, для которого разрабатывался данный проект. Изображения хранятся в разрешении 28x28 пикселей, в формате оттенков серого. Примеры изображений рукописных символов представлены на рисунке 9.

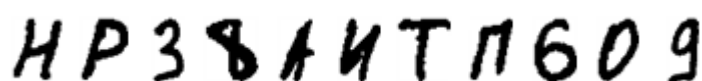


Рисунок 9 – Пример рукописных изображений

В ходе экспериментов по оценке производительности библиотек использована следующая тестовая инфраструктура: Ubuntu 12.04, Intel Core i5-3210M @ 2.5GHz (CPU эксперименты).

Вычислительные эксперименты проводились с помощью сверточной нейронной сети, состоящей из следующих слоев: convolution (in filters: 1, out

filters: 28, size: 5x5, stride: 1x1), pooling (size: 3x3, stride: 3x3), convolution (in filters:28, out filters: 56, size: 5x5, stride 1x1), pooling (size: 2x2, stride: 2x2), FC.

Время обучения CNN, с помощью четырех рассмотренных библиотек, представлено на рисунке 10. Самую низкую производительность показала библиотека Pylearn2. Лучший результат, запущенных на CPU, показала библиотека Torch.

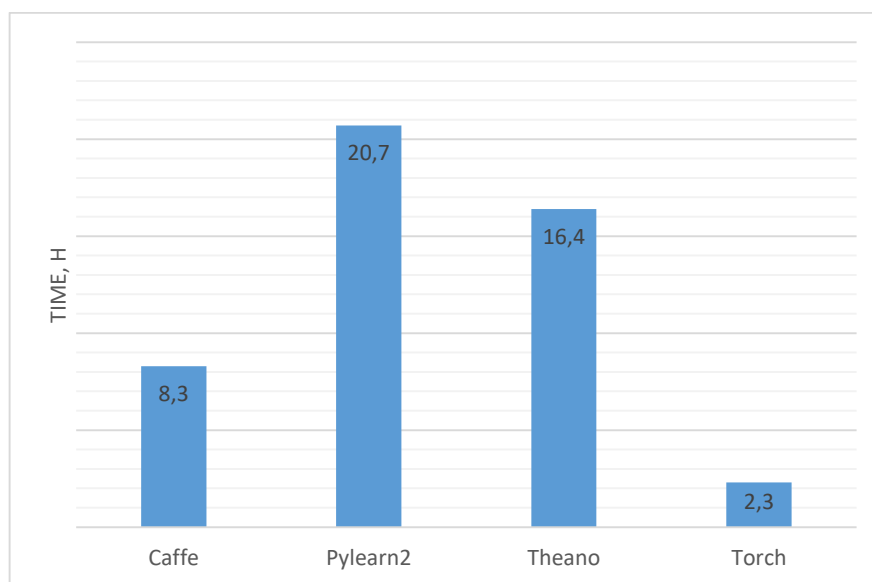


Рисунок 10 – Время обучения СНС

По времени классификации одного изображения на CPU с помощью обученных моделей, можно сделать вывод, что библиотека Torch оказалась вне конкуренции. Немного от нее отстала Caffe. Время классификации одного изображения представлено на рисунке 11.

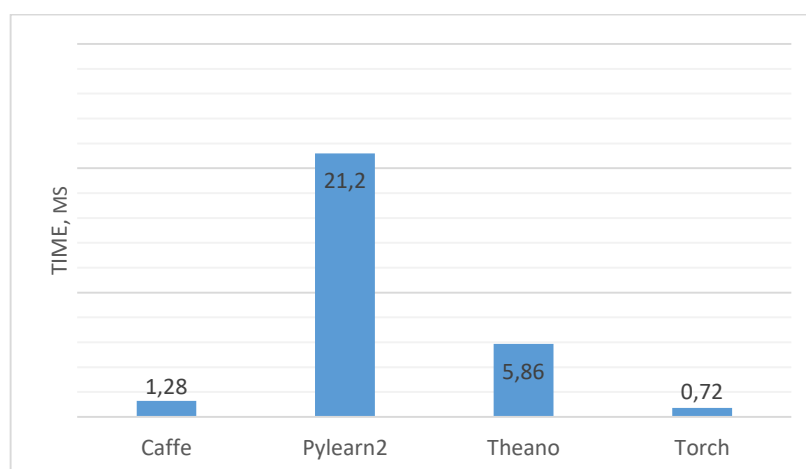


Рисунок 11 – Время классификации одного изображения с помощью обученных СНС

Точность классификации всех библиотек не превышает 97,4%. Полученные значения ниже приведенных на сайте MNIST для СНС. Такие отличия объясняются различиями в настройке начальных весов сетей и параметров методов оптимизации, применяемых в процессе обучения. Среднее значение точности классификации для каждой библиотеки представлено в таблице 2.

Таблица 2 – среднее значение точности классификации (5 экспериментов)

Библиотеки/ Вид сети	Caffe	Pylearn2	Theano	Torch
СНС	95,1%	96,3%	94,16%	97,4%

На основании проведенного исследования существующих библиотек и анализа производительности на примере распознавания рукописных символов, определена оценка по шкале от 1 до 3 по таким параметрам:

- Скорость обучения. Показывает время обучения СНС.
- Скорость классификации. Показывает время распознавания одного изображения.
- Удобство в использовании. Позволяет оценить время, потраченное на ознакомление с библиотекой.
- Гибкость настройки связей между слоями, установки параметров методов, а также наличие различных способов обработки данных.
- Объем функционала. Реализованы типовые методы глубокого обучения.
- Наличие и комфортное использования документации.

Оценки библиотек по каждому параметру приведены в таблице 3. Оценивание производилась по трех бальной шкале, чем выше балл, тем лучше соответствующий критерий библиотеки. При анализе библиотек было выявлено: Библиотека Caffe показала лучшую скорость обучения и оказалась самой комфортной в использовании. Со стороны функциональных



возможностей лучшие показатели были у Pylearn2. Наиболее гибкой оказалась библиотека Theano. Библиотека Torch имеет более ясную и детальную документацию, так же она имеет средние показатели по каждому критерию.

Таблица 3 – Оценки библиотек по заданным параметрам

Критерии/ библиотеки	Скорость обучения	Скорость классификации	Удобство	Гибкость	Функционал	Документация	Итоговая оценка
Caffe	3	2	3	1	1	2	12
Pylearn2	1	1	2	1	3	1	9
Torch	2	3	2	2	2	3	14
Theano	2	1	1	3	2	1	10

Более сформированной оказалась библиотека Torch. По скорости обучения, удобству не уступает ей библиотеки Caffe.

После анализа уже существующих библиотек появляется возможность выбора наиболее подходящей. С учетом операционной системы предприятия, для которого разрабатывался данный проект и языком программирования на котором написана разработка, такие библиотеки как Torch и Caffe не подходят. Оставшиеся библиотеки показали не лучшие результаты. Библиотека Pylearn2 имеет низкую скорость обучения и классификации, не гибкая, но главным недостатком данной библиотеки является ограниченность документации, что сильно усложнит работу с ней. Theano так же имеет низкую скорость классификации, не удобна в использовании и документация ограничена. Еще одним существенным минусом данных библиотек является трудность их применения к языку, на котором разрабатывался проект. В следствие всех вышеперечисленных недостатков, было принято решение создать собственную библиотеку нейросетевого распознавания рукописных символов, которая будет полностью соответствовать нуждам заказчика.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

В данной ИНС была использована архитектура «Обучение с учителем», которая способна распознавать множество бланков, в автоматическом режиме. Режим обучения данной ИНС реализована по эталонным ответам.

Для реализации программы была использована Visual Studio 2017, которая содержит подключенную и расширенную интегрированную среду разработки с инновационными функциями для повышения производительности разработчика, дополнительные инструменты с поддержкой разработки приложений для Windows 10, веб-разработки, а также улучшения средств отладки и оптимизации исполняемого и управляемого кода. В качестве языковой среды был выбран язык C#.

### 2.1 Обучающая выборка

В качестве одного из наборов, данных для обучения и тестирования сети будет использоваться база изображений рукописных цифр, предоставленная предприятием для которого осуществляется разработка. Изображения имеют разрешение 28x28 пикселей и хранятся в формате оттенков серого, следовательно, каждое значение пикселя лежит в диапазоне от 0 (представляет белый цвет) до 255 (представляет черный цвет). Символы отцентрированы на изображении. На рисунке 12 показаны десять изображений в обучающем наборе.

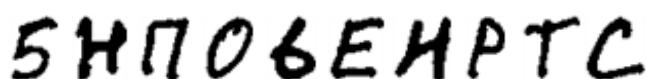


Рисунок 12 – Пример символов выборки

Для этих целей будет использоваться обучающая выборка, содержащая 27124 изображений рукописных символов, включающая в себя цифры и буквы русского алфавита. Частота распределения рукописных символов представлена в таблице 4.

Таблица 4 – Частота распределения рукописных символов

Символ	Количество	Символ	Количество	Символ	Количество
0	252	Е	753	У	15
1	1380	Ё	12	Ф	136
2	939	Ж	363	Х	132
3	1022	З	422	Ц	743
4	1034	И	532	Ч	238
5	852	Й	154	Ш	55
6	752	К	421	Щ	52
7	985	Л	421	Ъ	868
8	397	М	782	Ы	245
9	235	Н	489	Ь	149
А	1587	О	1590	Э	225
Б	910	П	1465	Ю	467
В	973	Р	968	Я	908
Г	1134	С	794		
Д	1134	Т	139		

Далее представлена гистограмма распределения рукописных символов, рисунок 13.

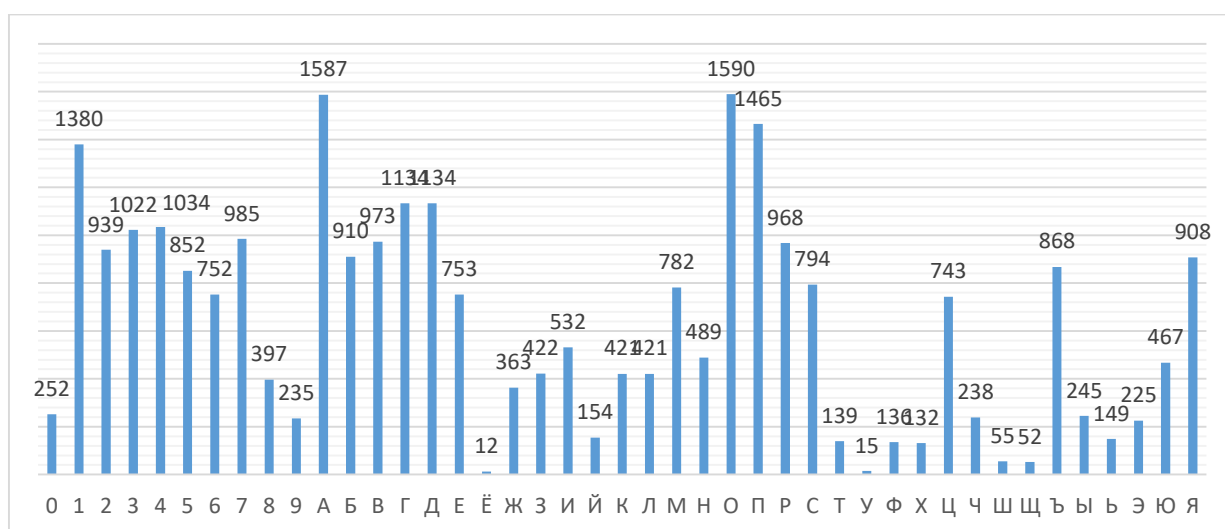


Рисунок 13 – Гистограмма распределения рукописных символов в выборке

Как можно видеть из рисунка 13, не смотря на большое количество обучающих примеров в исходном наборе, количество букв в выборке распределено не равномерно, следовательно, было принято решение создать обучающую выборку, содержащую около 100 – 200 примеров для каждого символа. Символы, которые не имели такого количества изображений были размножены за счет имеющихся изображений этих символов. Таблица 5, содержит количество изображений для каждого символа в получившейся выборке.

Таблица 5 – Количество символов в усредненной выборке

Символ	Количество	Символ	Количество	Символ	Количество
0	252	Е	214	У	211
1	220	Ё	211	Ф	136
2	239	Ж	225	Х	132
3	222	З	212	Ц	119
4	214	И	209	Ч	238
5	205	Й	209	Ш	100
6	210	К	204	Щ	105
7	213	Л	213	Ъ	100
8	237	М	207	Ы	210
9	235	Н	210	Ь	205
А	209	О	200	Э	151
Б	210	П	228	Ю	206
В	209	Р	214	Я	205
Г	213	С	210		
Д	210	Т	210		

Размер сбалансированной выборки составил – 8592 изображения. Изображения в данной базе имеют размер 28x28 пикселей и хранятся в формате оттенков серого, следовательно, каждое значение пикселя лежит в

диапазоне от 0 (черный цвет) до 255 (белый цвет), символы центрированы на изображении.

## 2.2 Определение оптимальных параметров для реализации CNN

Ниже приведены результаты тестирования сверточной сети для набора данных рукописных символов. Для задачи распознавания рукописных символов из обучающей выборки элементы матрицы входного изображения записываются по строкам. Для обучения сети использовались различные модификации метода градиентного спуска, коррекция весов производилась после 200 элементов из обучающей выборки, такой способ обучения сети, является оптимальным вариант между последовательным (стохастическим) и пакетным обучением. Влияние весов представлено в таблице 6.

Таблица 6 – Влияние количества эпох на точность распознавания

Размер мини выборки = 100				
Количество эпох	Алгоритмы оптимизации			
	SGD	Adam	AdaGrad	AdaDelta
5	81,04%	94,55%	96,14%	93,09%
10	84,52%	95,20%	96,20%	94,21%
15	86,45%	95, 31%	96, 22%	94,76%
20	88,42%	95,97%	96,25%	95,82%

На рисунке 14, показан график изменение точности распознавания сети на проверочные выборки. Проанализировав данный график можно сделать вывод что количество эпох равное 20 является избыточным, так как значительное изменение точности происходит с 1 по 10 эпоху, а после 10 эпохи значение точности начинают колебаться, что свидетельствует о переобучении сети. Следовательно, для дальнейших экспериментов следует ограничить количество эпох обучение, не более 10.

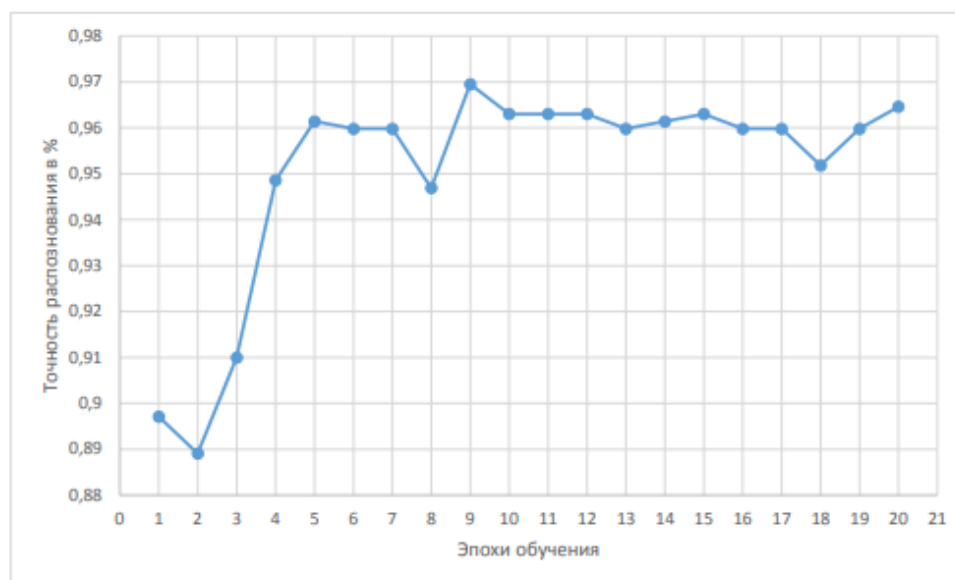


Рисунок 14 – Изменение точности распознавания для тестовой выборки

Таблица 7 – Влияние размера минивыборки на точность распознавания

Количество эпох = 10				
Размер мини выборки	Алгоритмы оптимизации			
	SGD	Adam	AdaGrad	AdaDelta
100	89,24%	95,97%	96,19%	95,08%
150	84,52%	95,20%	96,20%	94,21%
200	83,93%	94,55%	96,25%	94,22%

В таблице 6 приведены значения точности для разного количества эпох, лучшие результаты были получены для размера минивыборки 200, учитывая это, при обучении предпочтительней использовать размер минивыборки = 200.

Исходя из данных таблиц 6 и 7, можно сделать вывод что уменьшение размера мини выборки и увеличения количества эпох незначительно влияет на точность распознавания. Следовательно, дальнейшая точность распознавания может быть увеличена либо за счёт увеличения карт признаков в сверточном слое, либо за счет увеличения глубины сети.

Таблица 8 – Влияние количества карт признаков на точность распознавания

Количество эпох =10, размер мини выборки = 100				
Количество карт признаков	Алгоритмы оптимизации			
	SGD	Adam	AdaGrad	AdaDelta
32	84,42%	95,20%	96,20%	94,22%
64	85,23%	95,42%	96,30%	95,98%

Увеличение карт признаков в сверточном слое не привело к значительному изменению точности распознавания, следовательно, можно попробовать увеличить точность за счет добавления сверточных и подвыборочных слоев. Архитектура сверточной сети описана в таблице 9.

Таблица 9 – Архитектура сверточной нейронной сети

Слой	1	2	3	4	5	6
Тип слоя	Свертки	Пуллинг	Свертка	Пуллинг	Полносвязный	Полносвязный
Количество карт признаков	24x24 Ядра: 5x5	12x12 Ядра: 2x2	8x8 Ядра: 5x5	4x4 Ядра: 2x2	16 нейронов	64 нейронов
Функция активации	ReLU	-	ReLU	-	ReLU	Softmax

В таблице 10 приведены результаты тестирования описанной выше архитектуры сверточной сети.

Таблица 10 – Влияние размера минивыборки на точность распознавания

Количество эпох= 10				
Размер мини выборки	Алгоритмы оптимизации			
	SGD	Adam	AdaGrad	AdaDelta
100	84,73%	95,46	96,43%	95,96%
200	56,72%	95,03	96,30%	94,81%

Максимальная точность работы на тестовом наборе данных составила 96.43%, исходя из этого можно сделать вывод, что в условиях небольшого количества обучающих данных достигнутая точность в 96.43% является максимально доступной для данного объема обучающего набора.

В качестве оптимального решения была выбрана архитектура сверточной сети со следующими параметрами обучения:

- Алгоритм оптимизации: AdaGrad
- Количество эпох обучения: 10
- Размер минивыборки: 200

После тестирования сверточной нейронной сети на распознавание рукописных символов, так же было произведено тестирование распознавания рукописных слов. Результаты приведены в таблице 11.

Таблица 11 – Влияние размера выборки на точность распознавания

Количество эпох= 10				
Размер мини выборки	Алгоритмы оптимизации			
	SGD	Adam	AdaGrad	AdaDelta
100	74,34%	91,12%	94,30%	90,23%
200	46,72%	89,03%	94,56%	85,81%

Из Таблицы 11 видно, что такие алгоритмы, как SGD и AdaDelta сильно проседают в точности распознавания рукописных слов. Лучше всех проявил себя алгоритм AdaGrad показав точность распознавая 94,56%.

## 2.3 Проектирование логической модели системы

Построение логической модели начинается с определения классов и методов разрабатываемой системы. На рисунке 15 представлены основные классы системы.



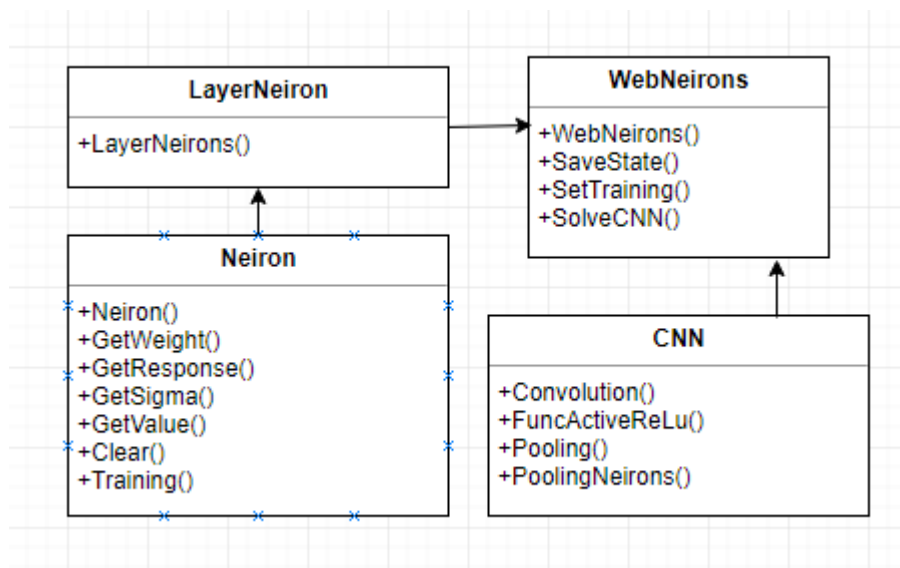


Рисунок 15– Модель СНС в виде UML диаграммы классов

*Класс Neiron* – содержит в себе функционал для нейрона, хранит в себе массив, который способен обучаться и сравнивать значения с уже содержащимися в памяти.

*Класс LayerNeurons* – описывает работу слоев нейронной сети.

*Класс WebNeurons* – представляет из себя хранилище массива нейронов. Подгружает его при создании из файла и сохраняет при выходе.

*Класс CNN* – содержит в себе функцию активации ReLU, реализацию свертки и пулинга (pooling).

В таблице 12 представлены основные методы вышеперечисленных классов.

Таблица 12 – Основные методы классов

Метод класса	Описание метода
<i>Класс Neiron</i>	
Neiron	Экземпляр нейрона.
Clear	Очищает память нейрона и присвоить ему новое имя.
GetResponse	Возвращает сумму величин отклонения входного массива от эталонного.

Метод класса	Описание метода
GetSigma	Производит подсчет суммы перемножения всех весов на сигму след слоя.
GetWeight	Обновляет значение весов для метода backpropagation.
GetValue	Производит подсчет значения текущего нейрона.
Training	Добавляет входной образ в память массива. Осуществляет проверку существования массива и соразмерен ли он массиву памяти.
Класс <i>LayerNeurons</i>	
LayerNeurons	Экземпляр слоя нейронов.
Класс <i>WebNeurons</i>	
WebNeurons	Экземпляр нейросети.
SaveState	Сохраняет всю нейронную сеть в файл.
SetTraining	Обучает нейроны с именем trainingName, получая новый вариант образа.
SolveCNN	Производит подсчет максимума из всех нейронов сети, возвращая имя нейрона.
Класс <i>CNN</i>	
Convolution	Реализует алгоритм Свертки.
FuncActiveReLU	Реализует работу функции активации ReLU
Pooling	Осуществляет работу алгоритма Pooling над массивом.
PoolingNeurons	Осуществляет работу алгоритма Pooling над слоями нейросети.

### 3 РЕЗУЛЬТАТЫ ПРОВЕДЕННОГО ИССЛЕДОВАНИЯ

Результаты распознавания рукописных символов были выявлены на основе расчета отношения верно распознанных символ к их общему числу, формула 18. Наилучшая точность распознавания рукописных символов составила 96,3%.

Формула расчета точности распознавания:

$$R = \frac{n}{N} \quad (18)$$

где  $R$  – точность распознавания по всему набору тестовой выборки,  
 $n$  – количество правильно распознанных символов из тестовой выборки,  
 $N$  – количество элементов в тестовой выборке.

Полученная точность имеет различие с достигнутой при тестировании библиотеки. Такое расхождение связано с настройкой некоторых гиперпараметров в ходе реализации модели. Далее приведены некорректно распознанные символы. Некорректное распознавание происходит за счет индивидуального почерка. Например, на рисунке 16 можно заметить, что буква «Д» соответствует параметрам буквы «А».

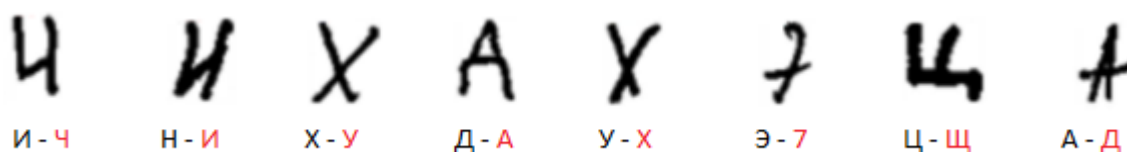


Рисунок 16 – Некорректно распознанные символы

Так же была рассчитана точность распознавания рукописных слов, которая составила 94,56%.

## **4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

В настоящее время для решения многих практических задач в различных сферах человеческой деятельности широкое распространение получили нейронные сети. Данные модели искусственного интеллекта используются при диагностике, прогнозировании, в решении оптимизационных задач, а также в задачах распознавания образов и анализа изображений.

Технико-экономическое обоснование научно-исследовательских работ проводится с целью определения и анализа трудовых и денежных затрат, направленных на их реализацию, а также уровня их научно-технической результативности.

В данном разделе необходимо решить следующий ряд задач: определить целевой рынок разработанного продукта, произвести анализ конкурентных решений, QuaD-анализ, SWOT-анализ. Так же рассчитать материальных затрат и определить ресурсоэффективность.

### **4.1 Предпроектный анализ**

#### **4.1.1 Потенциальные потребители результатов исследования**

Целевым рынком данной разработки являются организации, которым необходимо распознать рукописные символы на машиночитаемых бланках.

В первую очередь целевым предприятием является учебные учреждения, которые могут использовать данную разработку с целью проверки экзаменационных бланков. Например, бланков ЕГЭ.

Карта сегментирования рынка приведена в таблице 11.

Таблица 11 – Карта сегментирования рынка

Факторы	Сегменты рынка по группам потребителей						
	Шко-лы	Вузы	Научные центры	Мелкий бизнес	Крупная корпорация	Разработки	
						Theano	Caffe
Технические характеристики	1	2	3	3	3	3	2
Цена	3	2	3	3	0	1	2
Надежность	3	3	3	2	2	2	2
Удобство в использовании	3	1	2	1	3	1	3
Совместимость с оборудованием	3	2	2	3	3	2	1
Гибкость	1	2	2	2	3	3	1
Документация	3	2	1	2	2	1	2

Карта сегментирования рынка показывает, какие ниши не заняты конкурентами и где уровень конкуренции низок. Оценка степени значимости факторов разработки осуществляется по четырёхпольной шкале – от 0 до 3. Нулевой уровень означает, что фактор не имеет значения для данного сегмента потребителей. Максимальное значение равно 3, оно показывает, что данный фактор имеет особое значение для сегмента. Например, цена со значение 3 свидетельствует о том, что этот фактор наиболее значим и потребитель заинтересован в минимально возможной цене.

Из карты сегментирования рынка представленной в таблице 11 можно определить соответствие существующих разработок и требования потребителей, приобретающих данный продукт. Например, библиотека Caffe почти соответствует требованиям потребителей, приобретающих данный продукт для использования в школах, а библиотека Theano больше подходит для крупных корпораций. Соответствия выявленные при составлении карты

сегментирования рынка показывают, что существующие разработки нуждаются в доработке.

Новая разработка будет востребованной при повышении уровня надежности, совместимости с оборудованием и программного обеспечения. Так же разработанный продукт должен иметь минимальную цену, вследствие чего разработка станет доступной таким потребителям, как школы, вузы и научные центры.

#### 4.1.2 Анализ конкурентных технических решений

В качестве конкурентов целесообразно рассмотреть такие библиотеки как: Theano, Caffe. Экспертная оценка основных технических характеристик данных продуктов представлена в таблице 12.

Таблица 12 – Оценочная карта сравнения конкурентных систем

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Повышение производительности труда пользователя	0,12	5	4	4	0,6	0,48	0,48
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,12	5	5	3	0,6	0,6	0,48
3. Надежность	0,12	5	5	5	0,6	0,6	0,6
4. Безопасность	0,08	5	5	5	0,4	0,4	0,4
5. Потребность в ресурсах памяти	0,12	4	3	3	0,48	0,48	0,36
6. Функциональная мощность (предоставляемые возможности)	0,11	4	3	4	0,44	0,44	0,33
7. Простота эксплуатации	0,08	5	5	3	0,4	0,24	0,32
8. Качество интеллектуального интерфейса	0,08	4	4	3	0,32	0,32	0,32

Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,06	5	4	5	0,3	0,24	0,3
2. Уровень проникновения на рынок	0,04	3	4	3	0,12	0,16	0,12
3. Цена	0,01	5	5	5	0,05	0,05	0,05
4. Послепродажное обслуживание	0,05	5	4	3	0,25	0,2	0,15
5. Срок выхода на рынок	0,01	4	5	4	0,04	0,05	0,04
<b>Итого</b>	<b>1</b>				<b>4,6</b>	<b>4,26</b>	<b>3,95</b>

Из проведенного сравнения библиотек нейросетевого распознавания следует, что уязвимость конкурентных технических решений связана, прежде всего, с удобством в эксплуатации и уровнем проникновения на рынок.

#### 4.1.3 QuaD-анализ

Для описания качества новой разработки и ее перспективности на рынке использована технология QuaD. Оценка разработки, с учетом ее технических и экономических особенностей, создания и коммерциализации, представлена в таблице 13.

Таблица 13 – QuaD-анализ

Критерии оценки	Вес критерия	Баллы	Макс. балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
1.Повышение производительности и труда пользователя	0,12	95	100	0,95	0,114
2.Удобство в эксплуатации (соответствует требованиям потребителей)	0,12	90	100	0,9	0,108
3.Надежность	0,12	90	100	0,9	0,108

Критерии оценки	Вес критерия	Баллы	Макс. балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
4.Безопасность	0,08	95	100	0,95	0,076
5.Потребность в ресурсах памяти	0,12	85	100	0,85	0,102
6.Функциональная мощность (предоставляемые возможности)	0,11	85	100	0,85	0,0935
7.Простота эксплуатации	0,08	90	100	0,9	0,072
8.Качество интеллектуального интерфейса	0,08	80	100	0,8	0,064
<b>Показатели оценки коммерческого потенциала разработки</b>					
9.Конкурентоспособность продукта	0,06	95	100	0,95	0,057
10.Уровень проникновения на рынок	0,04	70	100	0,7	0,028
11.Цена	0,01	100	100	1	0,01
12.Послепродажное обслуживание	0,05	100	100	1	0,05
13.Срок выхода на рынок	0,01	85	100	0,85	0,0085
<b>Итого</b>	<b>1</b>				<b>89,1%</b>

В результате проведенного анализа перспективность разработки равна 89,1%. Поскольку значение показателя входит в промежуток от 80 до 100, данная разработка является перспективной.



#### 4.1.4 SWOT-анализ

SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Анализ заключается в описании сильных и слабых сторон проекта, в выявлении возможностей и угроз для реализации проекта, которые проявились или могут появиться в его внешней и внутренней среде. В результате составлена итоговая матрица SWOT-анализа, которая приведена в таблице 14:

Таблица 14 – Итоговая матрица SWOT

	<b>Сильные стороны проекта:</b>  С1. Высокая точность распознавания символов  С2. Наличие дополнительного функционала, облегчающего работу пользователей  С3. Открытый исходный код  С4. Высокая скорость обработки  С5. Широкая область применения	<b>Слабые стороны проекта:</b>  Сл1. Необходимость обучения искусственной нейронной сети  Сл2. Высокая потребность ресурсов памяти  Сл3. Отсутствие кроссплатформенности  Сл4. Сильная зависимость от оборудования
<b>Возможности:</b>  В1. Добавление новых функциональных возможностей с учетом пожеланий заказчика  В2. Реализация новых способов взаимодействия с другими модулями системы  В3. Заказ подобной разработки другими заказчиками  В4. Улучшение интерфейса	<b>Направления развития:</b>  1. В1С3С5 – Открытый исходный код и широкая область применения позволят добавить новые функциональные возможности.  2. В2С2С3С5 – наличие дополнительного функционала, открытого кода и широкой области применения позволит реализовать новые способы взаимодействия с другими модулями системы.	<b>Сдерживающие факторы:</b>  1. В1Сл2 Сл4 – добавление новых функциональных возможностей может быть невозможным из-за отсутствия ресурсов памяти и сильной зависимости от оборудования.  2. В2Сл2Сл3Сл4 - высокая потребность ресурсов памяти, отсутствие кроссплатформенности и сильной зависимости от оборудования могут воспрепятствовать реализации новых способов

	<p>3.В3С1С2С3С4С5 – сильные стороны проекта позволят привлечь потенциальных заказчиков.</p> <p>4.В4С2С5 - наличие дополнительного функционала и широкая область применения позволят улучшить интерфейс.</p>	<p>взаимодействия с другими модулями системы.</p> <p>3. В3Сл1Сл2Сл3Сл4 – слабые стороны проекта могут воспрепятствовать появлению потенциальных заказчиков.</p>
<p><b>Угрозы:</b></p> <p>У1. Появление на рынке конкурентоспособных аналогов разработки</p> <p>У2. Несвоевременное финансирование</p> <p>У3. Неверное выполнение инструкций пользователем</p> <p>У4. Потеря актуальности</p>	<p>1. У1У4С1С2С4С5 - высокая точность распознавания символов, наличие дополнительного функционала, высокая скорость обработки и широкая область применения позволят избежать появления конкурентоспособных аналогов и потерю актуальности разработки</p>	<p>1.У1У4 Сл1Сл2Сл3Сл4 – слабые стороны проекта могут привести к появлению конкурентоспособных аналогов разработки и потери актуальности.</p>

На основе результатов итоговой матрицы можно сделать вывод, что основными недостатками данной разработки является отсутствие кроссплатформенности и сильная зависимость от оборудования. Наличие кроссплатформенности обеспечит продукту актуальность и конкурентоспособность на рынке. Так же уменьшение зависимости разработки от оборудования позволит реализовать новые способы взаимодействия с другими модулями системы.

## 4.2 Определение возможных альтернатив разработки

Для определения альтернативных вариантов реализации технической задачи используется морфологический подход. Морфологическая матрица для составляющих реализации рассматриваемого проекта представлена в таблице 15.

Таблица 15 – Морфологическая матрица

	1	2	3
А. Язык разработки	C#	Python	C++
Б. ОС	Windows	Linux	Mac
В. Фреймворк	.NET Framework	Jupyter Notebook	JUCE
Г. Тип ИНС	Полностью связанная нейросеть (fully connected neural network, FC NN)	Сверточные нейронные сети (convolutional neural network, CNN)	Автокодировщик (autoencoder, AE)

Из полученной морфологической матрицы, можно получить как минимум 3 варианта реализации проекта:

- Исполнение 1. A1B1B1Г2.
- Исполнение 2. A2B2B2Г1.
- Исполнение 3. A3B3B3Г3.

В рамках ВКР реализуется первый вариант исполнения, а два других необходимы для проведения сравнительного анализа.

### 4.3 Организация и планирование работ

В данном разделе составляется список проводимых работ, определяются их исполнители и продолжительность. Так как число исполнителей не превышает двух, линейный график работ является наиболее удобным и компактным способом представления данных планирования. График выполнения научно-исследовательской работы представлен в таблице 16.

Таблица 16 – Перечень этапов работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Постановка задач	1	Описание требований	РП
	2	Анализ предметной области	РП, С
	3	Разработка технического задания	РП, С
Проектирование	4	Разработка проектного решения	РП, С
Реализация	5	Разработка графического интерфейса для обучения ИНС	С
	6	Разработка слоёв свертки и линейной ректификации (ReLU)	С
	7	Разработка слоя пулинг	С
	8	Разработка слой полносвязного слоя	
	9	Реализация архитектуры нейронной сети	С
	10	Интеграция с подсистемой предобработки и сегментации	С
Тестирование	11	Функциональное тестирование	С
Внедрение	12	Внедрение	РП, С
Оформление документации	13	Оформление пояснительной записки	С
	14	Проверка работы	НР

Примечание к таблице 1.10: РП – руководитель от предприятия; НР — научный руководитель; С — студент.

### 4.3.1 Продолжительность этапов работ

Расчет продолжительности этапов работ может осуществляться опытно- статистическим методом. Для расчета ожидаемого значения продолжительности работ  $t_{ож}$  применяются две оценки:  $t_{min}$  и  $t_{max}$  (метод двух оценок).

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}, \quad (19)$$

где  $t_{min}$  – минимальная трудоемкость работ, чел/дн;

$t_{max}$  – максимальная трудоемкость работ, чел/дн.

Для выполнения перечисленных работ требуются специалисты: руководитель от предприятия (РП), научный руководитель (НР) и студент (С).

Для построения линейного графика рассчитывается длительность этапов в рабочих днях, а затем осуществляется её перевод в календарные дни. Расчёт продолжительности выполнения каждого этапа в рабочих днях ( $T_{рд}$ ) выполняется по формуле:

$$T_{рд} = \frac{t_{ож}}{K_{вн}} \cdot K_{д}, \quad (20)$$

где  $t_{ож}$  – продолжительность работы, дн.;

$K_{вн}$  – коэффициент выполнения работ ( $K_{вн} = 1$ );

$K_{д}$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ( $K_{д} = 1,2$ ).

Расчёт продолжительности этапа в календарных днях осуществляется по формуле:

$$T_{кд} = T_{рд} \cdot T_{к}, \quad (21)$$

где  $T_{кд}$  – продолжительность выполнения этапа в календарных днях;

$T_{рд}$  – продолжительность выполнения этапа в рабочих днях;

$T_{к}$  – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (22)$$

где  $T_{КАЛ}$  – календарные дни,  $T_{КАЛ} = 365$ ;

$T_{ВД}$  – выходные дни,  $T_{ВД} = 52$ ;

$T_{ПД}$  – праздничные дни,  $T_{ПД} = 10$ .

Подставив значения в формулу 22, получим следующий результат:

$$T_{КАЛ} = \frac{365}{365 - 52 - 10} = 1,205 \quad (23)$$

В таблице 17 приведена длительность этапов работ и число исполнителей, занятых на каждом этапе.

Таблица 17 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях			Длительность работ в календарных днях		
	tmin, чел-дни			tmax,чел-дни			toж, чел-дни											
	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
Описание требований	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1,2	1,2	1,2
Анализ предметной области	5	5	5	10	10	10	7	7	7	2	2	2	3,5	3,5	3,5	4,2	4,2	4,2
Разработка ТЗ	8	8	8	16	16	16	11	11	11	2	2	2	5,5	5,5	5,5	6,6	6,6	6,6
Разработка проектного решения	2	5	5	4	10	10	3	7	7	2	2	2	1,5	3,5	3,5	1,8	4,2	4,2
Разработка графического интерфейса для обучения ИНС	2	3	3	4	6	6	3	4	4	1	1	1	3	4	4	3,6	4,8	4,8
Разработка слоёв свертки и ReLu	25	27	28	50	54	56	35	38	39	1	1	1	35	38	39	42,2	45,8	47
Разработка слоя пулинга	11	20	25	18	40	50	14	28	35	1	1	1	14	28	35	16,9	33,7	42,2
Разработка слой полносвязного слоя	15	20	20	25	40	40	19	28	28	1	1	1	19	28	28	22,9	33,7	33,7
Реализация архитектуры ИНС	8	17	18	16	34	36	11	24	25	1	1	1	11	24	25	13,3	28,9	30,1
Интеграция с подсистемой предобработки и сегментации	2	4	4	4	8	8	3	6	6	1	1	1	3	6	6	3,6	7,2	7,2
Функциональное тестирование	3	3	3	6	6	6	4	4	4	1	1	1	4	4	4	4,8	4,8	4,8
Внедрение разработанной библиотеки	1	1	1	2	2	2	1	1	1	2	2	2	0,5	0,5	0,5	0,6	0,6	0,6
Оформление пояснительной записки	16	16	16	32	32	32	22	22	22	1	1	1	22	22	22	26,5	26,5	26,5
Проверка работы	6	6	6	12	12	12	8	8	8	1	1	1	8	8	8	9,6	9,6	9,6
Итого							142	189	198				131	176	185	158	211,8	222,7

Для наглядного отображения графика и распределения работ между участниками проекта использована диаграмма Ганта (таблица 18).

Таблица 18 – Календарный план-график проведения работ

№ раб	Содержание работ	Должность исполнителя	$T_{ki}$	Продолжительность выполнения работ, декады.																	
				декабрь			январь			февраль			март			апрель			май		
				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Описание требований	РП	1,2	■																	
2	Анализ предметной области	РП, С	4,2	■	■																
3	Разработка ТЗ	РП, С	6,6		■	■															
4	Разработка проектного решения	РП, С	1,8			■															
5	Разработка графического интерфейса	С	3,6			■															
6	Разработка слоёв свертки и ReLu	С	42,2				■	■	■	■	■	■									
7	Разработка Polling-слоя	С	16,9							■	■	■									
8	Разработка полносвязного слоя	С	22,9									■	■	■	■						
9	Реализация архитектуры ИНС	С	13,3										■	■	■						
10	Интеграция с подсистемой предобработки и сегментации	С	3,6												■						
11	Функциональное тестирование	С	4,8												■						
12	Внедрение разработанной библиотеки	РП, С	0,6													■					
13	Оформление пояснительной записки	С	26,5													■	■	■	■	■	■
14	Проверка работы	НР	9,6																	■	■

■

 Студент (С)

■

 Научный руководитель (НР)

■

 Руководитель от предприятия (РП)



### 4.3.2 Расчет накопления технической готовности

В данном разделе производится оценка текущих результатов работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего этапа выполнен общий объем работ по проекту в целом.

Степень готовности определяется формулой:

$$СГ_i = \frac{ТР_i^H}{ТР_{общ}} = \frac{\sum_{k=1}^i ТР_k}{ТР_{общ}} = \frac{\sum_{k=1}^i \sum_{j=1}^m ТР_{km}}{\sum_{k=1}^I \sum_{j=1}^m ТР_{km}}, \quad (24)$$

где  $ТР_{общ}$  – общая трудоемкость проекта;

$ТР_i$  ( $ТР_k$ ) – трудоемкость  $i$ -го ( $k$ -го) этапа проекта,  $i = \overline{1, I}$ ;

$ТР_i^H$  – накопленная трудоемкость  $i$ -го этапа проекта по его завершении;

$ТР_{ij}$  ( $ТР_{kj}$ ) – трудоемкость работ, выполняемых  $j$ -м участником на  $i$ -м этапе.

Таблица 19 – Нарастание технической готовности работы для И1

Этап	$ТР_i$ , %	$СГ_i$ , %
1. Описание требований	0,76	0,76
2. Анализ предметной области	2,66	3,42
3. Разработка ТЗ	4,18	7,6
4. Разработка проектного решения	1,14	8,75
5. Разработка графического интерфейса для обучения ИНС	2,28	11,03
6. Разработка слоёв свертки и ReLu	26,74	37,77
7. Разработка слоя пулинга	10,71	48,48
8. Разработка слой полносвязного слоя	14,51	62,99
9. Реализация архитектуры ИНС	8,43	71,42
10. Интеграция с подсистемой предобработки и сегментации	2,28	73,7
11. Функциональное тестирование	3,04	76,74

Этап	ТР <sub>i</sub> , %	СГ <sub>i</sub> , %
12.Внедрение разработанной библиотеки	0,38	77,12
13.Оформление пояснительной записки	16,79	93,92
14.Проверка работы	6,08	100

#### 4.4 Бюджет научно-технического исследования

В состав бюджета входит стоимость всех расходов, необходимых для выполнения работ по проекту. При формировании бюджета используется группировка затрат по следующим статьям:

- материальные затраты;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

##### 4.4.1 Расчет материальных затрат

Данная статья включает стоимость всех материалов, используемых при разработке проекта. Для выполнения проекта использован один персональный компьютер в компании. Мелкие расходы (канцелярия, затраты на печать и пр.) могут быть отнесены к статье прочих расходов.

Таблица 20 – Материальные затраты

Наименование	Ед. измерения	Количество			Цена за ед., руб.			Затраты на материалы, (З <sub>м</sub> ), руб.		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Персональный компьютер	шт	1	1	1	2500 0	2500 0	2500 0	2500 0	2500 0	2500 0
Итого								2500 0	2500 0	2500 0

#### 4.4.2 Расчет основной заработной платы исполнителей системы

Данная статья расходов включает заработную плату научного руководителя и студента, а также премии и доплаты. Расчет основной заработной платы выполняется на основе трудоёмкости выполнения каждого этапа и величины месячного оклада исполнителя.

Величина месячного оклада научного руководителя (МОНР) получена из открытых данных, размещенных на официальном сайте Национального исследовательского Томского политехнического университета. Величина месячного оклада инженеров (МОИ) берется как месячный оклад инженера кафедры.

Основной расчет фонда заработной платы выполняется по формуле:

$$ЗП_{\text{дн-т}} = MO/N, \quad (25)$$

где МО – месячный оклад, руб.;

$N$  – количество рабочих дней в месяц, при шестидневной рабочей неделе –  $N=24,91$ , а при пятидневной рабочей неделе –  $N=20,58$ .

Среднедневная заработная плата научного руководителя равна:

$$ЗП_{\text{дн-т}} = \frac{33264,9}{24,91} = 1335,4 \frac{\text{руб.}}{\text{раб.день}}, \quad (26)$$

Среднедневная заработная плата руководителя от предприятия равна:

$$ЗП_{\text{дн-т}} = \frac{41000}{24,91} = 1645,9 \frac{\text{руб.}}{\text{раб.день}}, \quad (27)$$

Среднедневная заработная плата студента равна:

$$ЗП_{\text{дн-т}} = \frac{7250}{24,91} = 291 \frac{\text{руб.}}{\text{раб.день}}, \quad (28)$$

Расчет основной заработной платы приведен в таблице 21.

Таблица 21 – Основная заработная плата исполнителей системы

Исполнитель	Оклад, руб./мес	Среднедневная ставка	Затраты времени, раб. дни			Коэффициент	Фонд з/платы, руб.		
			Ис п. 1	Ис п. 2	Ис п. 3		Исп. 1	Исп. 2	Исп. 3
Научный руководитель	33264,9	1335,4	8	8	8	1,3	10683,2	10683,2	10683,2
Руководитель от предприятия	41000	1645,9	23	27	27	1,3	37855,7	44439,3	44439,3
Студент	7250	291	133	180	189	1,3	38703	52380	54999
Итого:							87241,9	107502,5	110121,5

#### 4.4.3 Расчет затрат по дополнительной заработной плате

По данной статье предусмотрены затраты по дополнительной заработной плате, учитывающие величину доплат, связанных с обеспечением гарантий и компенсаций. Расчёты дополнительной заработной платы приведены в таблице 22.

Таблица 22 – Дополнительная заработная плата исполнителей системы

Исполнитель	Основная заработная плата, руб.			Коэффициен т	Дополнительная заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3		Исп. 1	Исп. 2	Исп. 3
Руководитель	10683, 2	10683, 2	10683, 2	0,12	1281,98	1281,9 8	1281,98
Администрато р	37855, 7	44439, 3	44439, 3		4542,68	5332,7 2	5332,72
Студент	38703	52380	54999		4644,36	6285,6	6599,88
Итого					10469,0 3	12900, 3	13214,5 8

#### 4.4.4 Расчет отчислений во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Таблица 23 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель	10683,2	10683,2	10683,2	1281,98	1281,98	1281,98
Администратор	37855,7	44439,3	44439,3	4542,68	5332,72	5332,72
Студент	38703	52380	54999	4644,36	6285,6	6599,88
Коэффициент ПФРФ	0,22					
Коэффициент ФСС	0,029					
Коэффициент ФФОМС	0,051					
Итого						
Исполнение 1	29313,28					
Исполнение 2	36120,84					
Исполнение 3	37000,82					

#### 4.4.5 Расчет накладных расходов

Накладные расходы учитывают все затраты, не вошедшие в предыдущие статьи расходов: печать и ксерокопирование, оплата электроэнергии, оплата пользования услугами и пр. Величину коэффициента накладных расходов можно принять в размере 16%.

Для исполнения 1:  $Z_{\text{накл}} = 152024,2 * 0,16 = 24323,9$  рублей.

Для исполнения 2:  $Z_{\text{накл}} = 181523,6 * 0,16 = 29043,8$  рублей.

Для исполнения 3:  $Z_{\text{накл}} = 185336,9 * 0,16 = 29653,9$  рублей.

#### 4.4.6 Формирование бюджета затрат проекта

Сумма затрат по всем статьям расходов рассчитывается и заносится на данном этапе в таблицу 24.

Таблица 24 – Бюджет затрат проекта

Статья затрат	Сумма, руб.		
	Исп. 1	Исп. 2	Исп. 3
Материальные затраты НТИ	25000,0	25000,0	25000,0
Основная заработная плата исполнителей темы	87241,9	107502,5	110121,5
Дополнительная заработная плата исполнителей темы	10469,028	12900,3	13214,58
Отчисления во внебюджетные фонды (страховые отчисления)	29313,28	36120,84	37000,82
Накладные расходы	24323,9	29043,8	29653,9
Итого	176348,11	210567,44	214990,8

На основании данных таблицы 1.17 можно сделать вывод о том, что наименьший бюджет для реализации работ приходится на 1 исполнение. Первый вариант исполнения является наиболее выгодным в плане реализации.

#### 4.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования. Определение эффективности производится путем определения интегрального показателя эффективности научного исследования через нахождение величин финансовой и ресурсной эффективности.

Интегральный финансовый показатель определяется по следующей формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}} \quad (29)$$

где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

Исполнение 1:  $I_{\text{финр}} = 176348,11 / 214990,8 = 0,82$ ;

Исполнение 2:  $I_{\text{финр}} = 210567,44 / 214990,8 = 0,98$ ;

Исполнение 3:  $I_{\text{финр}} = 214990,8 / 214990,8 = 1$ .

Интегральный показатель ресурсоэффективности определяется по формуле:

$$I_{pi} = \sum_i^n a_i b_i, \quad (30)$$

где  $I_{pi}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта исполнения разработки;

$a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;

$b_i$  – бальная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

$n$  – число параметров сравнения.

Бальная оценка каждого варианта исполнения по техническим критериям представлена в таблице 25.

Таблица 25 – Сравнительная оценка характеристик вариантов исполнения  
проекта

Критерии	Объект исследования			
	Весовой коэффициент	Исп.1	Исп.2	Исп.3
1.Повышение производительности труда пользователя	0,14	5	5	3
2.Удобство в эксплуатации (соответствует требованиям потребителей)	0,14	5	5	3
3.Надежность	0,14	5	5	4
4.Безопасность	0,103	5	5	5
5.Потребность в ресурсах памяти	0,14	4	3	3
6.Функциональная мощность (предоставляемые возможности)	0,131	4	4	4
7.Простота эксплуатации	0,103	5	3	3
8.Качество интеллектуального интерфейса	0,103	4	4	4
<b>Ресурсоэффективность:</b>		<b>4,63</b>	<b>4,28</b>	<b>3,58</b>

Интегральный показатель эффективности вариантов исполнения разработки определяется по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{исп.1}^{финр}} \quad (31)$$

После этого определяется сравнительная эффективность исполнений разработки, которая позволит определить самый выгодный вариант разработки с позиции финансовой и ресурсной эффективности:

$$\mathcal{E}_{cp} = \frac{I_{исп.1}}{I_{исп.2}} \quad (32)$$

Сравнительная эффективность разработки приведена в таблице 26.



Таблица 26 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0,82	0,98	1
2	Интегральный показатель ресурсоэффективности разработки	4,63	4,28	3,58
3	Интегральный показатель эффективности	5,64	4,37	3,58
4	Сравнительная эффективность вариантов исполнения	1	0,77	0,64

По результатам таблицы 26 можно сделать вывод, что самым эффективным исполнением с позиции ресурсоэффективности и финансовой эффективности является первое исполнение и его стоимость равна 176348,11.

Второе исполнение занимает второе место. Интегральный показатель эффективности на 22% ниже первого, а сумма бюджета составляет 210567,44.

Наименее эффективным является третье исполнение, его интегральный показатель эффективности на 13% ниже второго исполнения, сумма бюджета 214990,8.

#### 4.6 Вывод

В данном разделе был определен основной рынок, на который направлена разработанная библиотека, а именно мелкие школы, крупные высшие учебные заведения, мелкие научно-исследовательские центры.

Проведение QuaD – анализа доказало перспективность разработки. Так же были определены преимущества и недостатки разработанной библиотеки по отношению к конкурирующим разработкам. Анализ возможных альтернатив разработки, на основе морфологической матрицы, показал три возможные исполнения. Для определения этапов работы и продолжительности был составлен список проводимых работ. Длительность работы в календарных днях составила 158. Для наглядного распределения

работ между участниками проекта, построена диаграмма Ганта. Определены сметы затрат на выполнение проекта. Финансовые затраты на первое исполнение составили 176348,11 рублей. Разница по сравнению со вторым исполнением составляет 34219,33 рубля. Проведен анализ сравнительной эффективности исполнений. Наиболее эффективной оказалось первое исполнение.

В результате можно сделать вывод что выбор первого исполнение является полностью оправданным как с финансовой, так и с точки зрения эффективности.

## **5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

Выполнение работы заключалось в разработке библиотеки нейросетевого распознавания рукописных символов на машиночитаемых бланках. Разработка программы велась при помощи компьютера. Сферы применения разработки: от нужд обычных пользователей до специализированных задач. Одним из наиболее перспективных применений - обработка машиночитаемых бланков ЕГЭ. Взаимодействие пользователя с разработанной программой производится с помощью программных и аппаратных средств ПЭВМ и периферийных устройств, подключенных к ним.

Данный раздел содержит описание производственной и экологической безопасности, а также безопасности в чрезвычайных ситуациях. Кроме того, рассмотрены правовые и организационные вопросы обеспечения безопасности на рабочем месте в процессе реализации проекта.

### **5.1 Производственная безопасность**

Длительная работа с ПЭВМ приводит к возникновению вредных, опасных производственных факторов. Под вредным фактором понимается негативное воздействие на человека, приводящее к ухудшению самочувствия или заболеваниям [12]. Согласно ГОСТ 12.0.003-74 все вредные и опасные факторы, возникающие в процессе трудовой деятельности, можно разделить на физических, химические, биологические и психофизиологические. В таблице 27 представлены результаты анализа вредных и опасных факторов, возникающих в процессе работы над проектом.

Таблица 27 – Опасные и вредные факторы при выполнении работы за  
ПЭВМ [13]

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПЭВМ	1.Превышение показателей электромагнитного излучения; 2.Отклонение показателей микроклимата; 3.Превышение уровней шума и вибрации; 4.Недостаточная освещенность рабочей зоны; 5.Психофизические факторы.	1.Поражение электрическим током; 2.Возникновение пожара	1)СанПиН 2.2.4.548-96; 2)СанПиН 2.2.2/2.4.1340-03; 3)СП 52.13330.2011; 4)ГОСТ Р 12.1.019-2009 ССБТ; 5) СНиП 21-01-97

## 5.2 Вредные производственные факторы

### 5.2.1 Повышенный уровень электромагнитных излучений

Выполнение работ осуществлялось в офисном помещении с постоянно работающих ПЭВМ, основными источниками электромагнитных полей являлись токоведущие части данных ПЭВМ [13].

При работе компьютера вокруг него образуется электромагнитное поле, деионизирующее окружающую среду, что делает воздух сухим, слабо ионизированным. Согласно СанПиН 2.22.542-96 напряженность электромагнитного поля на расстоянии 50 см вокруг монитора по электрической составляющей должна быть в диапазоне частот: 5 Гц ÷ 2 кГц – 25 В/м и 2 кГц ÷ 400кГц – 2,5 В/м. Плотность магнитного потока должна быть не в диапазоне частот: 5 Гц ÷ 2 кГц – 250 нТл и 2 кГц ÷ 400кГц – 25 нТл.

Возможные способы защиты от ЭМИ: использование жидкокристаллических мониторов, максимально отдаление оператора от монитора, сокращение времени работы за компьютером и увеличение количества перерывов, применение ионизаторов воздуха.

## 5.2.2 Повышенные показатели микроклимата

Микроклимат помещения – это комплекс метеорологических условий в данном помещении. По степени физической тяжести, работа программиста относится к категории лёгких работ по Сан Пин № [2.2.4.548-96]. В соответствии с временем года и категорией тяжести работ определены параметры микроклимата согласно требованиям, СанПиН 2.2.4.548 – 96 и приведены в таблице 28.

Таблица 28 – Оптимальные и допустимые нормы микроклимата производственных помещений ПЭВМ [14].

Период года	Температура, °С					Относительная влажность, %		Скорость движения воздуха, м/с	
	Оптим- альная	Допустимая на рабочих местах				Оптив - альная	Допу- стим- ая	Опти- мальн -ая, не более	Допус - тимая, не более
		Верхняя		Нижняя					
		Пост.	Не пост.	Пост.	Не пост.				
Холодный	22 – 24	25	26	21	18	40 – 60	75	0,1	0,1
Теплый	23 – 25	28	30	22	20	40 – 60	70	0,1	0,1

В помещении, где производится разработка, температура и влажность воздуха поддерживается в заданных в таблице пределах. Кроме того, имеется автоматическая система кондиционирования, очищающая и нагревающая (охлаждающая) поступающий в кабинет воздух. Таким образом, нет необходимости в принятии дополнительных мер для создания благоприятных условий.

### 5.2.3 Повышенный уровень шума

Работе в условиях повышенного шума и вибрации может привести к: ухудшению слуха или к его полной потере, возникновению головной боли, снижению концентрации внимания и памяти, вследствие чего понижается производительность труда. Основными источниками шума при работе с ПК являются составные части работающего ПК: вентилятор на процессоре, жесткий диск. Стандарт 2.2.4/2.1.8.562 – 96 распространяется на технологическое оборудование, машины и другие источники шума, которые создают в воздушной среде все виды шумов. Предельно допустимые нормы по госту 12.1.003-83 уровня шума для рабочих мест приведены в таблице 29.

Таблица 29 – Допустимые уровни звукового давления и уровня звука на рабочих местах по госту 12.1.003-83

Рабочие места	Уровни звукового давления, дБ, в активной полосе частот, Гц								Уровни звука, дБ
	63	125	250	500	1000	2000	4000	8000	
А	71	61	54	49	45	42	40	38	50

А – помещение конструкторских бюро, лаборатории для теоретических работ.

Способы снижения шума: звукоизоляция помещений смежных с шумным производственным участком, акустический экран, средства индивидуальной защиты, чистка вентилятора от пыли или его замена на более новый.

### 5.2.4 Недостаточность освещенности рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни

которого регламентируются СП 52.13330.2011. Причиной являются недостаточность естественного освещения, искусственного освещения, пониженная контрастность. Недостаточность освещения снижает производительность труда, увеличивает утомляемость, количество допускаемых ошибок, ухудшению зрения. В таблице 30 представлены нормативные показатели искусственного освещения при работах заданной точности.

Таблица 30 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ.

Характеристики зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристики фона	Искусственное освещение		
						Освещённость, лк		
						При системе комбинированного освещения		При системе общего освещения
						Всего	от общего	
Высокой точности	От 0,3 до 0,5	III	г	Средний, большой	Светлый, средний	400	200	200

### 5.2.5 Психофизические факторы

Основную часть времени разработчик-программист проводит за работой на персональном компьютере. Рабочее место должно быть максимально удобным, т.к. в результате неправильной позы могут возникнуть проблемы со здоровьем. Поэтому следует соблюдать требования, описанные в документе СанПиН 2.2.2/2.4.1340-03: расстояние от экрана монитора до глаз работника должно составлять 60 - 70 см; экран монитора должен быть антибликовым, каждые 2 часа необходимо делать перерывы по 20 минут, работа за компьютером не должна превышать 6 часов.

Несоблюдение вышеуказанных правил может привести к получению работником травмы или развития заболевания.

## **5.3 Опасные производственные факторы**

### **5.3.1 Опасность поражения электрическим током**

Поражение электрическим током является опасным производственным фактором, поскольку оператор ПЭВМ имеет дело с электрооборудованием. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности: перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей, при обнаружении неисправности оборудования и приборов необходимо, вызвать мастера ответственного за оборудование [15,16].



### **5.3.2 Опасность возникновения пожара**

Возникновение пожара является опасным производственным фактором. Пожар наносит большой материальный ущерб, сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за следующих факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха, неисправности электрооборудования и освещения, неправильная их эксплуатация. Возможные виды источников воспламенения (искры): при разряде статического электричества, от электрооборудования, от удара и трения [17].

### **5.4 Экологическая безопасность**

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

Разработка и работа за ПЭВМ не являются экологически опасными работами, потому объект, на котором производилась разработка продукта и объекты, на которых он будет использован операторами ПЭВМ относятся к предприятиям пятого класса, размер селитебной зоны для которых равен 50 м [19].

Разработанный проект магистерской диссертации, не наносит вреда окружающей среде ни на стадиях его разработки, ни на стадиях эксплуатации. Однако, средства, необходимые для его разработки и эксплуатации могут наносить вред окружающей среде. Например, аккумуляторы. Для их утилизации необходимо обращаться в специальные организации, занимающиеся приемом, утилизацией и переработкой.

## **5.5 Безопасность в чрезвычайных ситуациях**

Наиболее вероятной чрезвычайной ситуацией является возникновение пожара, так как на рабочем месте располагается большое количество ЭВМ. В соответствии с нормами пожарной безопасности, помещения с ЭВМ относятся к категории В (пожароопасные).

Основные причины возникновения возгораний: нарушение правил эксплуатации электрического оборудования, эксплуатация его в неисправном состоянии, перегрузка электрических сетей, применение неисправных осветительных приборов, электропроводки и устройств [18].

Для предупреждения возгораний в помещении необходимо соблюдать: установленный режим эксплуатации электрических устройств, противопожарные нормы и правила при установке оборудования, проводить технические осмотры и планово-предупредительные ремонты оборудования и технических средств противопожарной защиты и пожаротушения (огнетушители) согласно утвержденному графику.

В помещении должен быть установлен углекислотный огнетушитель типа ОУ-5 для тушения пожаров.

## **5.6 Правовые и организационные вопросы обеспечения безопасности**

При разработке проектных решений в вычислительных центрах не обязателен режим сокращенного рабочего дня, достаточно установление перерывов в работе. Работа в вычислительном центре относится к классу 2 – допустимые условия труда, при которых уровень факторов среды и трудового процесса, не превышает уровней, установленных гигиеническими нормативами для рабочих мест. В связи с этим дополнительных льгот и компенсаций работникам этой области не предусмотрено.

Органы, регулирующие соблюдение федерального законодательства и нормативных правовых актов: федеральная инспекция труда, государственная экспертиза условий труда Федеральная служба по труду и занятости населения (Минтруда России Федеральная служба по экологическому, технологическому и атомному надзору (Госгортехнадзор, Госэнергонадзор, Госатомнадзор России) Федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека (Госсанэпиднадзор России)).

Общественный экологический контроль проводится профсоюзными и общественными организациями и объединениями.

### **Организационные мероприятия обеспечения безопасности рабочей зоны**

При выполнении выпускной квалификационной работы основная нагрузка приходится на центральную нервную систему, так как происходит умственная работа. При проектировании необходимо организовать комфортные условия для полноценной работы. По нормам СанПиН 2.2.2/2.4.1340 – 03 на протяжении рабочего дня должны быть регламентированы перерывы для качественной работы [15]. Время перерывов в течение рабочей смены устанавливается с учетом её длительности, вида и категории трудовой деятельности.

## ЗАКЛЮЧЕНИЕ

Распознавание рукописного текста является одной из важнейших задач при обработке машиночитаемых бланков. В качестве метода машинного обучения были выбраны нейронные сети.

В данной работе был проведен сравнительный анализ использования сверточной и простой нейронной сети, типа персептрон для распознавания рукописных символов. В результате проведенных исследований было установлено, что использования сверточной нейронной сети будет наиболее оптимальным для распознавания рукописного текста. Преимуществом данной сети является меньшее количество настраиваемых параметров и более высокая точность распознавания символов.

Так же выявлена наиболее подходящая архитектура СНС. Проанализированы алгоритмы оптимизации на основе метода градиентного спуска. Выбрана функция активации нейронов. Произведено тестирование уже существующих библиотек проводились с использованием базы данных рукописных символов предоставленном предприятием, для которого был разработан данный продукт.

Итогом проделанной работы является разработка библиотеки, включающая в себя основные классы и функции необходимые для реализации распознавания рукописных символов на машиночитаемых бланках. Точность распознавания рукописных символов составила 96,4%.

## СПИСОК ЛИТЕРАТУРЫ

1. Хайкин С. Нейронные сети: полный курс. М.: Вильямс, 2006. - 1104 с.
2. Маркелов А.А. Алгоритмы и программная система классификации полутоновых изображений на основе нейронных сетей: Диссертация на соискание ученой степени кандидата технических наук / А. А. Маркелов. – Томск, 2007.
3. Солдатова О.П., Гаршин А.А. Применение сверточной нейронной сети для распознавания рукописных цифр. Компьютерная оптика. – 2010. – Том 34, №2. – с. 252-260 – ISSN 0134-2452.
4. В.Г. Спицын, Интеллектуальные системы: учебное пособие / В.Г. Спицын, Ю.Р. Цой; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2012. – 176 с
5. LeCun, Y. Efficient BackProp in Neural Networks: Tricks of the trade / Y.LeCun, L. Bottou, G. Orr, K. Muller – Springer, 1998.
6. LeCun, Y. Scaling learning algorithms towards AI / Y.LeCun, Y. Bengio – MITPress, 2007
7. Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Стакан, 206. – 53 с.
8. Amari, Shunichi. "A theory of adaptive pattern classifiers."Electronic Computers, IEEE Transactions on 3 (1967): 299-307.
9. Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization."The Journal of Machine Learning Research 12 (2011): 2121-2159.
10. Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method."arXiv preprint arXiv:1212.5701 (2012).
11. Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization."arXiv preprint arXiv:1412.6980 (2014)

12. Охрана труда. Основы безопасности жизнедеятельности// [www.Grandars.ru](http://www.Grandars.ru). 2016. [Электронный ресурс] Точка доступа: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohranatruda.html> (дата обращения: 19.05.2018).

13. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. – 2-е изд., перераб. и доп. – М.: Издательство «Альфа-Пресс», 2008. – 176 с.

14. СанПиН 2.2.4.548-96. Санитарные правила и нормы. Гигиенические требования к микроклимату производственных помещений // Библиотека гостей и нормативов. 2016. [Электронный ресурс] Точка доступа: [http://ohranatruda.ru/ot\\_biblio/normativ/data\\_normativ/5/5225/](http://ohranatruda.ru/ot_biblio/normativ/data_normativ/5/5225/) (дата обращения: 18.05.2018).

15. СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-вычисли- тельным машинам и организации работы // Библиотека гостей и нормативов. 2016. [Электронный ресурс] Точка доступа: [http://www.ohranatruda.ru/ot\\_biblio/normativ/data\\_normativ/39/39082/#i7287](http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/39/39082/#i7287) (дата обращения: 20.05.2018).

16. ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты // Электронный фонд правовой и нормативно- технической документации. 2010. [Электронный ресурс] Точка доступа: <http://docs.cntd.ru/document/gost-r-12-1-019-2009-ssbt> (дата обращения: 18.05.2017).

17. СНиП 21-01-97. Пожарная безопасность зданий и сооружений // Библиотека гостей и нормативов. 2016. [Электронный ресурс] Точка доступа: [http://www.ohranatruda.ru/ot\\_biblio/normativ/data\\_normativ/2/2107/](http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/2/2107/) (дата обращения: 19.05.2018).

18. Назаренко О. Б. Безопасность жизнедеятельности: учебное пособие / О. Б. Назаренко, Ю. А. Амелькович; Томский политехнический

университет. – 3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2013. – 178 с.

19. СанПиН 2.2.1/2.1.1.1200-03. Санитарно-эпидемиологические правила и нормативы. Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и других объектов // Библиотека гостов и нормативов. 2016. [Электронный ресурс] Точка доступа: [http://ohranatruda.ru/ot\\_biblio/normativ/data\\_normativ/11/11774/](http://ohranatruda.ru/ot_biblio/normativ/data_normativ/11/11774/) (дата обращения: 20.05.2018).

**ПРИЛОЖЕНИЕ А**  
**(обязательный)**

**RECOGNITION ALGORITHMS REVIEW**

Студент:

Группа	ФИО	Подпись	Дата
8ИМ6Б	Шалаева Алёна Александровна		

Консультант ОИТ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Мирошниченко Евгений Александрович	к.т.н.		

Консультант-лингвист ОИЯ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Диденко Анастасия Владимировна	канд. филол. наук, доц.		

Tomsk – 2018



## RECOGNITION ALGORITHMS REVIEW

This review presents the main points regarding neural networks, a method for training networks with backpropagation. To date, neural networks have been successfully applied to solve problems of classification or clustering of multidimensional data.

### 1 Artificial neural network, ANN

The ANN was created as a result of attempts to create biological neural networks. The ANN was used in the diagnosis, forecast, solving optimization problems, for pattern recognition and image analysis. The neural network has a high approximating ability. ANN is trained by using experimental data and it is able to solve the tasks posed with a high degree of accuracy. ANN is the dynamic system which consists of associated with each other formal neurons.

#### 1.1 The model of formal neuron

The model of formal neuron is a structural and functional part of the ANN. Such neuron is the elementary calculator (processor), it is used in the nodes of an ANN. The model of formal neuron is represented in Figure 1.

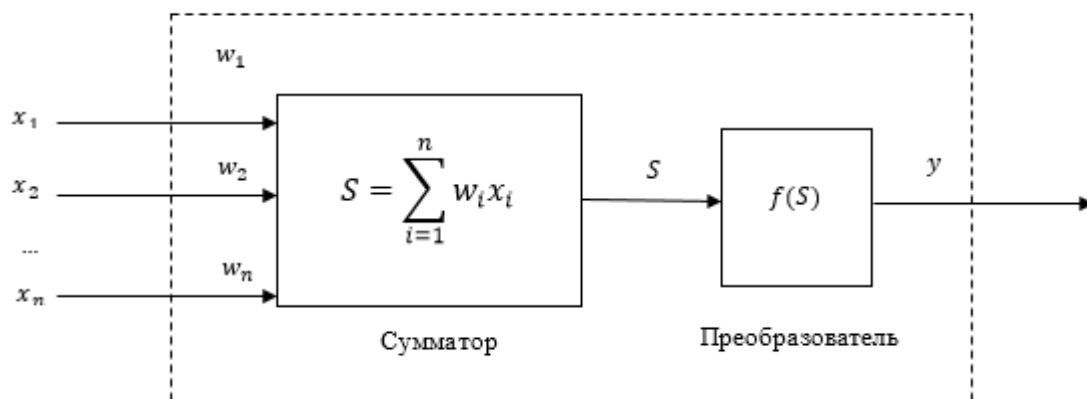


Figure 1 – Model of formal neuron

The neuron gets the input signals through a linear communication – the synapse. Synapse has a weight of connection between two neurons, it simulate throughput in the biological neurons. The next important element NNS is the adaptive adder. The main task of the adder is the summation of the input signals of the neurons. These signals are multiplied by the respective weight relations. The weighted sum is the result of the work of the adaptive adder, which characterizes the received signal of the neuron. The neuron must process this amount for generation of the output signal by using an activation function. Activation function accepts amount as an argument. As a result, the output signal is the value of the activation function.

## 1.2 Types of the activation function

The activation function has several main types, such as:

1. The function of a unit jump. The piecewise-constant function is equal to zero for negative values of the argument. The function is equal to one for positive values. At zero, this function is not defined. It is defined at this point by a certain number, so that the domain of the function definition contains all points of the real axis. Most often it does not matter that the function contains a value of zero.

$$f(S) = \begin{cases} 0, & S < 0 \\ 1, & S \geq 0 \end{cases}, \quad (1)$$

2. Sigmoid function is increased of monotonically differentiable S-shaped nonlinear function with saturation. The sigmoid amplifies weak signals and does not saturate with strong signals.

$$f(S) = \frac{1}{1 + e^{(-aS)}}, \quad (2)$$

3. Straightened linear activation function (ReLU)

The derivative of this function is equal to one or zero, so the gradients do not expand and attenuate. Because multiplying the one by the delta of the error is equal to the delta of the error. Using another function, the delta of the error could decrease or increase.

$$f(S) = \max(0, S) = \begin{cases} 0, & S < 0 \\ S, & S \geq 0 \end{cases} \quad (3)$$

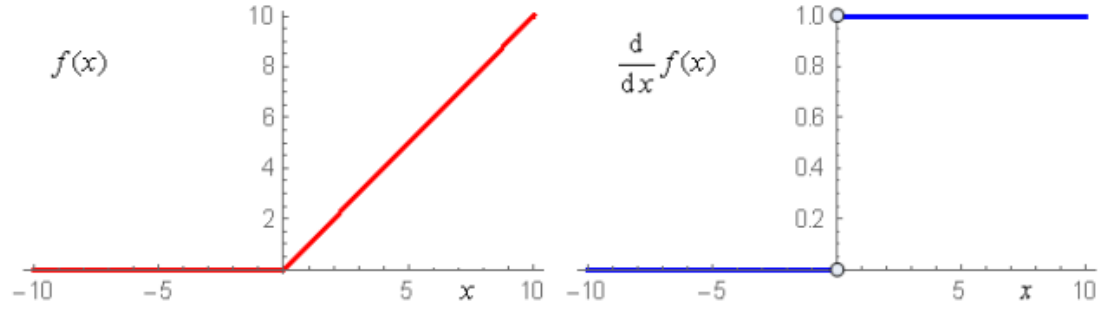


Figure 2 – Activation function ReLU

Advantages of the ReLU function include:

- 1) intensive action is not warranted;
- 2) unnecessary parts are removed;
- 3) the gradient does not change;
- 4) training is faster.

Disadvantages are:

- 1) Relu is not reliable in training;
- 2) Relu is dependent on the initialization of the weight.

4. Softmax function is a generalization of the logistic function for the multivariate case. The function converts a vector with real values into a vector of probabilities. The Softmax function is used in machine learning for classification tasks when the number of possible classes is greater than two (logistic function is used for two classes).

$$z_i = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)}, \quad (4)$$

The advantage of this function is based on the partial derivative of the  $i$ -th neuron:

$$\frac{\partial z_i}{\partial y_i} = z_i(1 - z_i), \quad (5)$$

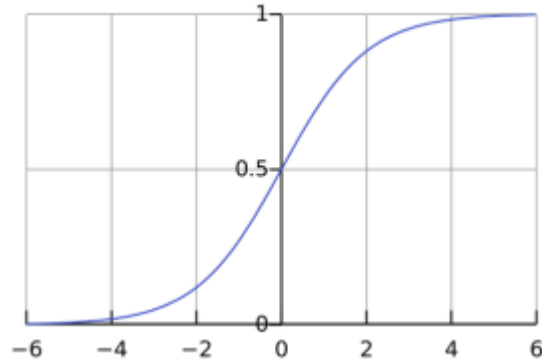


Figure 3 – Softmax function

### 1.3. Architecture of neural networks

ANN has a different architecture. The architectures of neural networks have common properties such as: firstly, they consist of formal neurons, and, secondly, the signal is produced using layers which in their turn are made up of neurons.

#### 1.3.1 Single-layer neural networks

In single-layer neural networks, signals from the input layer are fed to the output layer at once. It produces the necessary calculations, the results of which are immediately fed to the outputs.

#### 1.3.2 Multilayer neural networks

Such networks are also characterized by a hidden layer (layers). Covered layers are layers between the input and output layers. This structure of neural networks copies the multilayered structure of certain parts of the brain. Multilayered neural networks have much greater capabilities than single-layer networks. The work of hidden layers of neurons can be compared with the work of a large plant. The product (output signal) at the plant is collected by stages. After each machine, some intermediate result is obtained. Hidden layers also convert the input signals into some intermediate results.

### 1.3.3 Feedforward neural networks, FNN

Feedforward neural networks are classical architecture. The FNN (Figure 4) is a multilayer perceptron. The architecture has the following problems: firstly, many parameters. The network has many parameters, therefore, it is additionally retrained. As a result, the network can learn wrong and, secondly, the gradients fade away from the NN. There is small part is stayed from the gradient, in multilayer network. So weights do not change at the output.

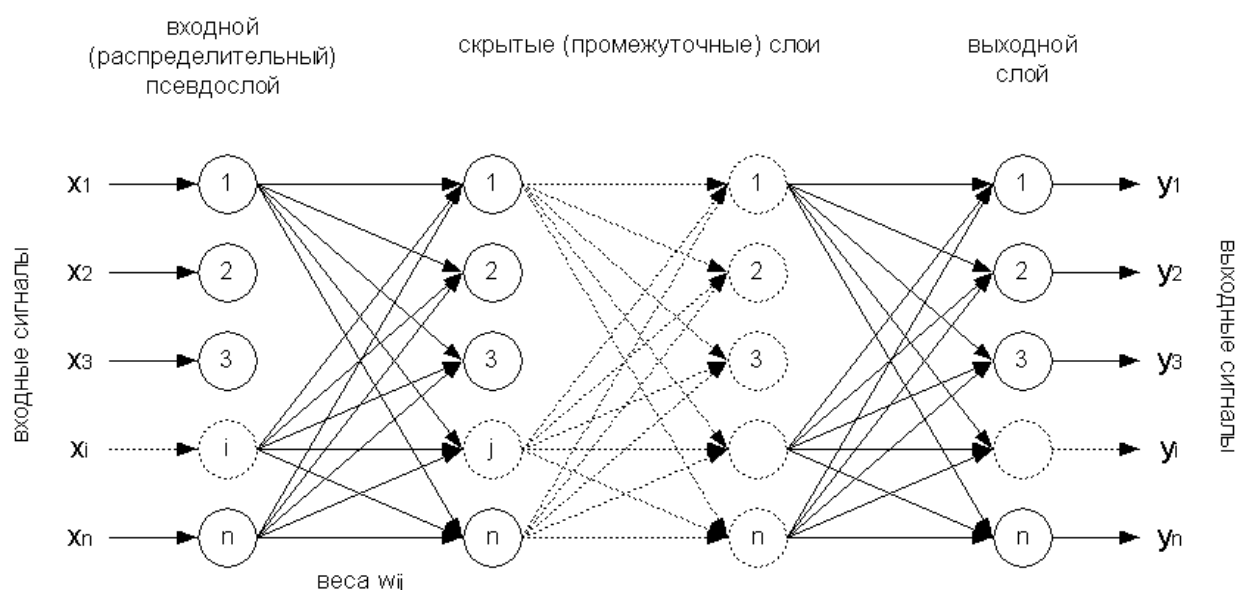


Figure 4 – Feedforward neural networks

### 1.3.4 Recurrent neural networks, RNN

The RNN has a cyclical joint. In the following step, the hidden layer forwards its own values to itself. The RNN realizes any activation function. Network has a certain type of memory and is better suited for working with sequences, context modeling, time dependencies. The model of such a network is shown in Figure 5.

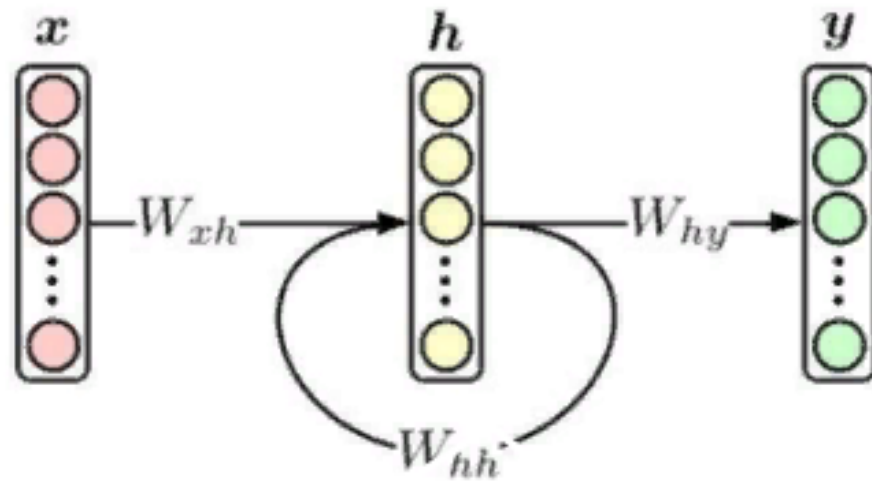


Figure 5 – Recurrent neural networks

### 1.3.5 Convolutional neural networks, CNN

Convolutional neural is the kind of network very different from the rest. CNN networks recognize objects located on the images. The network is the main component of deep training technology. The model of such a network is shown in Figure 6.

This architecture processes data in fragments, a sequential run is performed. Then the data moves further along the layers. In addition to the convolution layers, CNN uses the pool layers (P-pooling). The join layers are compressed with depth (usually a power of two). Several perceptrons (direct propagation network) are added to the final layers for the subsequent processing of data.

The CNN consists of, firstly, convolution layer and, secondly, pooling-layer and, thirdly, fully connected (FC).

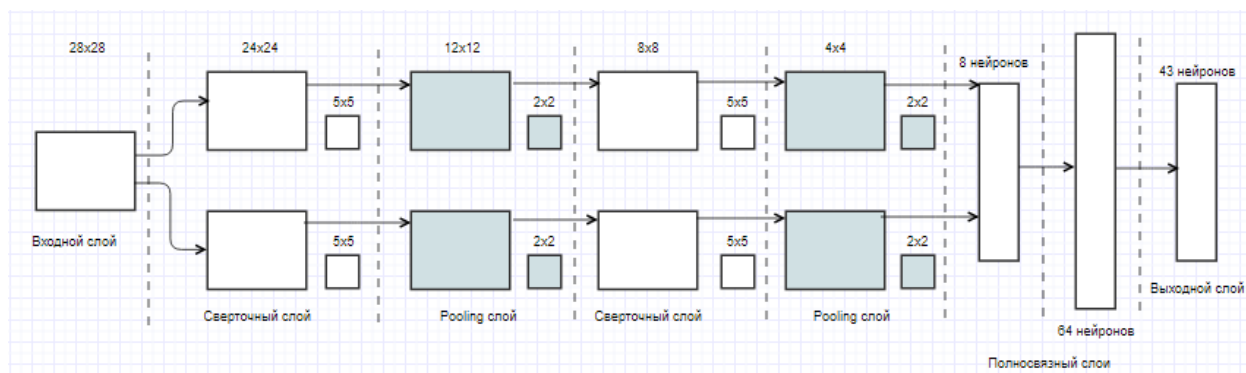


Figure 6 – Convolutional neural network

A convolutional layer can be called a matrix filter that changes the original image. This operation can be implemented several times. A layer consists of a matrix, called the core of convolution and image. The convolutional layer is implemented in this way: the core of the convolution is superimposed on the image fragment. The corresponding elements are multiplied. The obtained values are summed and fixed in the central location of the matrix.

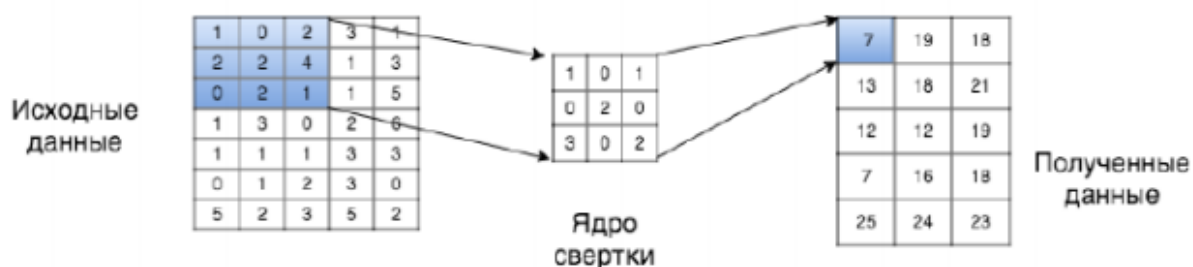


Figure 7 – Convolution layer

Pooling-layer reduces the measurement of the input characteristic map. Reducing the dimension can be effected by different ways. The most well-known method is the method of selecting the maximum element. This element is located around the current element.



Figure 8 – Pooling – layer

Multilayer perceptron covers the pooling and convolutional layers. The fully connected layer is effected by joining neurons of the current layer with neurons on the previous layer.

This architecture is most suitable for image recognition and classification. With its help, parallelization of computations is possible, as a consequence, the use of graphics processors. However, this architecture needs to be adjusted to a large number of variable parameters, such as the number of layers, the number of cores in each layer, the activation functions of each neuron, and many others.

#### 1.4 Algorithm for learning neural network

The most spread algorithm for learning neural networks is the method of back propagation of an error. The main idea: when the error is known and the derivative of the compressible function, the solution is effected in the opposite direction, also, weights of minimizing the error are recalculated.

The learning process of ANN is realized: input signals are examples from the training set. One such iteration is called the age. This learning process continues from era to era, until the weight of the neurons and the threshold levels of the ANN are stabilized. So the network error of the training set does not reach the specified minimum value.

The implementation is performed in the following ways:



1) Stochastic gradient descent (sequential mode) is one of the teaching tools of ANN. The weights of the neurons are corrected after the passage of each example of the training set according to the NC.

2) In batch mode, the modification in the weights of the links occurs only after the entry of the entire training sample of the same epoch into the input.

3) Mini-batch is a compromise between sequential and batch mode. The scales are corrected after the entrance to the part of the training sample of one epoch.

From the point of view of real-time processes, a sequential mode is more preferable than a batch mode, since it requires less internal storage for each synoptic connection. Moreover, by presenting training examples in a random order (in the process of successive correction of weights), the search in the space of weights can be made really stochastic. This, in turn, minimizes the possibility of stopping the algorithm at the point of any local minimum.

## ПРИЛОЖЕНИЕ Б

В данном приложении представлен листинг программы.

В данном примере представлен класс CNN (класс сверточной нейронной сети), реализующий операции сверточных нейронных сетей.

```
class CNN
{
    public static double FuncActiveReLU(double x) => Math.Max(0, x);

    public static LayerNeurons Convolution(LayerNeurons lay)
    {
        LayerNeurons layer = new LayerNeurons(lay.neurons2.Count);
        var wANDh = layer.KernelForNWeb.GetLength(0);
        double sum = 0d;

        int smechenie = ((int)(wANDh / 2)) * 2;
        for (int I = 0 + wANDh / 2; I < lay.neurons2.Count - smechenie; I++)
            for (int J = 0 + wANDh / 2; J < lay.neurons2[I].Count - smechenie; J++)
            {
                sum = 0d;
                for (int i = 0; i < wANDh; i++)
                    for (int j = 0; j < wANDh; j++)
                        sum += layer.KernelForNWeb[j, i] * lay.neurons2[I + i - 1][J
+ j - 1].znachenie;
                layer.neurons2[I][J].znachenie = FuncActiveReLU(sum);
            }

        int tempint = layer.neurons2.Count - smechenie;
        while (layer.neurons2.Count > tempint)
            layer.neurons2.RemoveAt(layer.neurons2.Count - 1);
        for (int i = 0; i < layer.neurons2.Count; i++)
            while (layer.neurons2[i].Count > tempint)
                layer.neurons2[i].RemoveAt(layer.neurons2[i].Count - 1);

        return PoolingNeurons(layer);
    }

    public static LayerNeurons PoolingNeurons(LayerNeurons layer)
    {
        int WAndH = layer.neurons2.Count;
        double max;
        for (int i = 0; i < WAndH; i += 2)
            for (int j = 0; j < WAndH; j += 2)
            {
                max = -1;
                if (max < layer.neurons2[i][j].znachenie) max =
layer.neurons2[i][j].znachenie;
                if (max < layer.neurons2[i+1][j].znachenie) max =
layer.neurons2[i+1][j].znachenie;
                if (max < layer.neurons2[i][j+1].znachenie) max =
layer.neurons2[i][j+1].znachenie;
                if (max < layer.neurons2[i+1][j+1].znachenie) max =
layer.neurons2[i+1][j+1].znachenie;
                layer.neurons2[i / 2][j / 2].znachenie = max;
            }

        int tempint = layer.neurons2.Count/2;
        while (layer.neurons2.Count > tempint)
            layer.neurons2.RemoveAt(layer.neurons2.Count - 1);
    }
}
```

```

        for (int i = 0; i < layer.neurons2.Count; i++)
            while (layer.neurons2[i].Count > tempint)
                layer.neurons2[i].RemoveAt(layer.neurons2[i].Count - 1);
        return layer;
    }
}

```

Далее представлена реализация класса Neuron.

```

public class Neuron//класс нейрона, нейрон хранит в себе массив определённого образ
{
    //он может обучаться и сравнивать значение с имеющимся в памяти
    public string name; //имя - текстовое значение образа который хранит нейрон
    public double[,] veight; //массив весов - именно это и есть память нейрона
    public int countTrainig; //кол-во вариантов образа в памяти для правильного
    пересчёта весов при обучении
    public double sdvig; //нейрон сдвига для каждого нейрона
    public double sigma; //для backpropagation
    public double znachenie; //значение нерона после функции активации

    public Neuron() { }
    Random rand = new Random();
    ///генерит от -0.5 до 0.5
    public double GetRandom() => //rand.Next(-50, 50) / 100.0; //от -0.5 до 0.5
    rand.Next(-100, 100) / 10000.0; //от -0.01 до 0.01

    public string GetName() { return name; }
    public void Clear(string name, int x, int y)
    {
        //очистить память нейрона и присвоить ему новое имя
        this.name = name;
        veight = new double[x, y];
        for (int n = 0; n < veight.GetLength(0); n++)
            for (int m = 0; m < veight.GetLength(1); m++)
                veight[n, m] = GetRandom();
        sdvig = GetRandom();
        countTrainig = 0;
    }
    /// <summary>
    /// подсчитывает значение текущего нейрона
    /// </summary>
    public void GetZnachenie(List<Neuron> data)
    {
        double zn = 0;
        int k = 0;
        for (int n = 0; n < veight.GetLength(0); n++)
            for (int m = 0; m < veight.GetLength(1); m++)
            {
                k = n * veight.GetLength(0) + m;
                zn += data[k].znachenie * veight[n, m];
                //k++;
            }
        this.znachenie = zn;
    }
    /// <summary>
    /// подсчитывает сумму перемножения всех весов на сигму след слоя
    /// </summary>
    public double GetSigma(List<Neuron> data, int index, int colcount)
    {
        double sig = 0;
        int n = index % colcount;
        int m = index / colcount;
        foreach (var neuron in data)
            sig += veight[n, m] * neuron.sigma;
        return sig;
    }
    /// <summary>

```

```

    /// обновление весов для backpropagation
    /// </summary>
    public void GetVeight(List<Neiron> data, double GiperN)
    {
        int k = 0;
        for (int p = 0; p < veight.GetLength(0); p++)
            for (int q = 0; q < veight.GetLength(1); q++)
            {
                veight[p, q] += -GiperN * sigma * data[k].znachenie;
                k++;
            }
    }
}

```

Реализация класса слоев нейронов для сверточной нейронной сети.

```

class LayerNeurons
{
    public const int neuronInArrayWidth = 28;
    public const int neuronInArrayHeight = 28;
    //public const int countLayersWithExit = 3;
    public int resOfLayer;
    public Neiron[,] neurons;
    public List<List<Neiron>> neurons2= new List<List<Neiron>>();

    public int resOfKernel = 5;
    public double[,] KernelForNWeb;
    public LayerNeurons(int res)
    {
        resOfLayer = res;
        neurons = new Neiron[resOfLayer, resOfLayer];
        for (int i = 0; i < res; i++)
            neurons2.Add(new List<Neiron>());
        for (int i = 0; i < res; i++)
            for (int j = 0; j < res; j++)
                neurons2[i].Add(new Neiron());

        KernelForNWeb = new double[resOfKernel, resOfKernel];
        Random rand = new Random();
        for (int i = 0; i < resOfKernel; i++)
            for (int j = 0; j < resOfKernel; j++)
                KernelForNWeb[i, j] = rand.Next(-50, 50) / 100.0;
    }
}

```

Класс нейронной сети, осуществляющий ее работу, например, со слоями.

```

class NW//класс хранилище массива нейронов neuronArray
{ //подгружает его при создании, из файла и сохраняет при выходе
    public const int neuronInArrayWidth = 28;
    public const int neuronInArrayHeight = 28;
    public static int[] resFullLayer = new int[2] { 6, 8 };
    public const int countLayersWithExit = 3;
    private const string memoryExit = "memoryExit.txt";//файл хранения сети выходного
    слоя
    private const string memoryHidden = "memoryHidden.txt";//файл хранения сети
    скрытых слоев
    public static List<List<Neiron>> ArrayOfLayer = new
    List<List<Neiron>>();//null//массив нейронов. Был "private List" стал public static
    double GiperA = 0.5;
    double GiperN = 0.5;
    double Etalon = 1;

    private double FuncActiveLogSigmoid(double x) => 1 / (1 + Math.Exp(-x * GiperA));
    private double FuncActiveReLU(double x) => Math.Max(0, x);
}

```

```

public NW()
{
    InitWebHiddenLayersFromFile();//инициализация скрытых слоев

    ArrayOfLayer.Add(InitWebExitLayer());//инициализация выходного слоя
}
private static void InitWebHiddenLayersFromFile()
{
    // функция открывает текстовый файл и преобразовывает его в массив нейронов
    List<Neuron> tempLayer;
    JavaScriptSerializer json;
    Object o;
    if (File.Exists(memoryHidden))
    {
        string[] lines = File.ReadAllLines(memoryHidden);
        if (lines.Length == 0) InitWebHiddenLayers();
        for (int i = 0; i < countLayersWithExit - 1; i++)
        {
            tempLayer = new List<Neuron>();
            for (int j = 0; j < neuronInArrayWidth * neuronInArrayHeight; j++)
            {
                string jStr = lines[i * (countLayersWithExit - 1) + j];
                json = new JavaScriptSerializer();
                o = json.Deserialize<Object>(jStr);
                tempLayer.Add(NCreate((Dictionary<string, Object>)o));
            }
            ArrayOfLayer.Add(tempLayer);
        }
    }
    else InitWebHiddenLayers();
}

private static void InitWebHiddenLayers()
{
    //инициализация скртых слоев
    List<Neuron> tempLayer = new List<Neuron>();
    Neuron neuron;
    for (int i = 0; i < countLayersWithExit - 1; i++)
    {
        tempLayer = new List<Neuron>();
        for (int j = 0; j < neuronInArrayWidth * neuronInArrayHeight; j++)
        {
            neuron = new Neuron { veight = new double[neuronInArrayWidth,
neuronInArrayHeight] };
            for (int n = 0; n < neuronInArrayWidth; n++)
                for (int m = 0; m < neuronInArrayHeight; m++)
                    neuron.veight[n, m] = neuron.GetRandom();
            neuron.sdvig = neuron.GetRandom();
            tempLayer.Add(neuron);
        }
        ArrayOfLayer.Add(tempLayer);
    }
}

private static List<Neuron> InitWebExitLayer()
{
    // функция открывает текстовый файл и преобразовывает его в массив нейронов
    if (!File.Exists(memoryExit)) return new List<Neuron>();
    string[] lines = File.ReadAllLines(memoryExit);
    if (lines.Length == 0) return new List<Neuron>();
    string jStr = lines[0];
    JavaScriptSerializer json = new JavaScriptSerializer();
    List<Object> objects = json.Deserialize<List<Object>>(jStr);
    List<Neuron> res = new List<Neuron>();
    foreach (var o in objects) res.Add(NCreate((Dictionary<string, Object>)o));
    return res;
}

private static void InitWebHiddenLayersFromFileCNN()
{
    // функция открывает текстовый файл и преобразовывает его в массив нейронов

```

```

List<Neuron> templayer;
JavaScriptSerializer json;
Object o;
if (File.Exists(memoryHidden))
{
    string[] lines = File.ReadAllLines(memoryHidden);
    if (lines.Length == 0) InitWebHiddenLayers();
    for (int i = 0; i < countLayersWithExit - 1; i++)
    {
        templayer = new List<Neuron>();
        for (int j = 0; j < neuronInArrayWidth * neuronInArrayHeight; j++)
        {
            string jStr = lines[i * (countLayersWithExit - 1) + j];
            json = new JavaScriptSerializer();
            o = json.Deserialize<Object>(jStr);
            templayer.Add(NCreate((Dictionary<string, Object>)o));
        }
        ArrayOfLayer.Add(templayer);
    }
}
else InitWebHiddenLayersCNN();
}
private static void InitWebHiddenLayersCNN()
{
    //инициализация скрытых слоев
    List<Neuron> templayer = new List<Neuron>();
    Neuron neuron;
    for (int i = 0; i < countLayersWithExit - 1; i++)
    {
        templayer = new List<Neuron>();
        if (i == 0)
        {
            for (int j = 0; j < resFullLayer[i] * resFullLayer[i]; j++)
            {
                neuron = new Neuron { veight = new double[resFullLayer[i],
resFullLayer[i]] };
                for (int n = 0; n < neuronInArrayWidth; n++)
                    for (int m = 0; m < neuronInArrayHeight; m++)
                        neuron.veight[n, m] = neuron.GetRandom();
                neuron.sdvig = neuron.GetRandom();
                templayer.Add(neuron);
            }
        }
        else
        {
            for (int j = 0; j < resFullLayer[i] * resFullLayer[i]; j++)
            {
                neuron = new Neuron { veight = new double[ArrayOfLayer[i -
1].Count, ArrayOfLayer[i - 1].Count] };
                for (int n = 0; n < neuronInArrayWidth; n++)
                    for (int m = 0; m < neuronInArrayHeight; m++)
                        neuron.veight[n, m] = neuron.GetRandom();
                neuron.sdvig = neuron.GetRandom();
                templayer.Add(neuron);
            }
        }
        ArrayOfLayer.Add(templayer);
    }
}
private static Neuron NCreate(Dictionary<string, object> o)
{
    // преобразовать структуру данных в класс нейрона
    Neuron res = new Neuron();
    res.name = (string)o["name"];
    res.countTrainig = (int)o["countTrainig"];
    Object[] veightData = (Object[])o["veight"];
}

```

```

        int arrSize = (int)Math.Sqrt(veightData.Length);
        res.veight = new double[arrSize, arrSize];
        int index = 0;
        for (int n = 0; n < res.veight.GetLength(0); n++)
            for (int m = 0; m < res.veight.GetLength(1); m++)
            {
                res.veight[n, m] = Double.Parse(veightData[index].ToString());
                index++;
            }
        return res;
    }
    public void SaveState()
    { //сохранить массив нейронов в файл
        JavaScriptSerializer json = new JavaScriptSerializer();//сохранение выходного
        string jStr = json.Serialize(ArrayOfLayer[countLayersWithExit - 1]);
        StreamWriter file = new StreamWriter(memoryExit);
        file.WriteLine(jStr);
        file.Close();

        file = new StreamWriter(memoryHidden);
        for (int i = 0; i < countLayersWithExit - 1; i++)//сохранение скрытых слоев
            foreach (var n in ArrayOfLayer[i])
            {
                json = new JavaScriptSerializer();
                jStr = json.Serialize(n);
                file.WriteLine(jStr);
            }
        file.Close();
    }
    public string[] GetLiteras()
    { // получить список имён образов, имеющихся в памяти
        var res = new List<string>();
        foreach (var n in ArrayOfLayer[countLayersWithExit - 1])
            res.Add(n.GetName());
        res.Sort();
        return res.ToArray();
    }
    public string Solve(int[,] arr)
    { //сравнить vx массив с каждым нейроном из сети и вернуть имя нейрона наиболее
        похожого на него
        for (int IndexOfLayer = 0; IndexOfLayer < countLayersWithExit;
            IndexOfLayer++)
            if (IndexOfLayer == 0)//первый слой особый случай
                foreach (var n in ArrayOfLayer[0])
                {
                    n.znachenie = 0;
                    for (int i = 0; i < arr.GetLength(0); i++)
                        for (int j = 0; j < arr.GetLength(1); j++)
                            n.znachenie += arr[i, j] * n.veight[i, j];
                    n.znachenie += n.sdvig;
                    n.znachenie = FuncActiveLogSigmoid(n.znachenie);
                }
            else
                foreach (var n in ArrayOfLayer[IndexOfLayer])
                {
                    n.GetZnachenie(ArrayOfLayer[IndexOfLayer - 1]);
                    n.znachenie += n.sdvig;
                    n.znachenie = FuncActiveLogSigmoid(n.znachenie);
                }

        string res = null;//поиск наиб совпадения(значения)
        double max = 0;

```

```

double d;
foreach (var n in ArrayOfLayer[countLayersWithExit - 1])
{
    d = n.znachenie;
    if (d > max)
    {
        max = d;
        res = n.GetName();
    }
}
return res;
}

public string SolveCNN(LayerNeurons layer)
{
    //сравнить вх массив с каждым нейроном из сети и вернуть имя нейрона наиболее
    //похожего на него

    for (int IndexOfLayer = 0; IndexOfLayer < countLayersWithExit;
        IndexOfLayer++)
    {
        if (IndexOfLayer == 0) //первый слой особый случай
        {
            foreach (var n in ArrayOfLayer[0])
            {
                n.znachenie = 0;
                for (int i = 0; i < layer.neurons2.Count; i++)
                {
                    for (int j = 0; j < layer.neurons2[i].Count; j++)
                    {
                        n.znachenie += layer.neurons2[i][j].znachenie *
n.veight[i, j];

                        n.znachenie += n.sdvig;
                        n.znachenie = FuncActiveLogSigmoid(n.znachenie);
                    }
                }
            }
        }
        else
        {
            foreach (var n in ArrayOfLayer[IndexOfLayer])
            {
                n.GetZnachenie(ArrayOfLayer[IndexOfLayer - 1]);
                n.znachenie += n.sdvig;
                n.znachenie = FuncActiveLogSigmoid(n.znachenie);
            }
        }

        string res = null; //поиск наиб совпадения(значения)
        double max = 0;
        double d;
        foreach (var n in ArrayOfLayer[countLayersWithExit - 1])
        {
            d = n.znachenie;
            if (d > max)
            {
                max = d;
                res = n.GetName();
            }
        }
        return res;
    }

    public void SetTraining(string trainingName, int[,] data)
    {
        //занести в память нейрона с именем trainingName новый вариант образа data
        Neuron neuron = ArrayOfLayer[countLayersWithExit - 1].Find(v =>
v.name.Equals(trainingName)); //находим нейрон с прав ответом, он особенный
        if (neuron == null) // если нейрона с таким именем не существует, создадим
        //новыи и добавим его в массив нейронов
        {
            neuron = new Neuron();
            neuron.Clear(trainingName, neuronInArrayWidth, neuronInArrayHeight);
            ArrayOfLayer[countLayersWithExit - 1].Add(neuron);
        }
    }
}

```



```

        foreach (var n in ArrayOfLayer[countLayersWithExit - 1]) { //расчет на
выходном слое, формула расчета иная чем на скрытых слоях
            if (n.Equals(neiron))
                Etalon = 1;
            else Etalon = 0;
            n.sigma = GiperA * n.znachenie * (1 - n.znachenie) * (n.znachenie -
Etalon);
        }
        for (int i = countLayersWithExit - 2; i >= 0; i--) //расчет сигм всех слоев
кроме выходного
        {
            int index = 0;
            foreach (var n in ArrayOfLayer[i]) //int j = 0; j < neiroweb[i].Count;
j++)
            {
                n.sigma = GiperA * n.znachenie * (1 - n.znachenie);
                n.sigma *= n.GetSigma(ArrayOfLayer[i + 1], index,
neironInArrayWidth);
                index++;
            }
        }
        //////////////////////////////////расчет весов////////////////////////////////////
        for (int i = countLayersWithExit - 1; i > 0; i--) //расчет всех слоев и
выходного
        {
            foreach (var n in ArrayOfLayer[i])
            {
                n.GetVeight(ArrayOfLayer[i - 1], GiperN);
                n.sdvig += -GiperN * n.sigma * 1;
            }
            foreach (var n in ArrayOfLayer[0]) //расчет на первом слое
            { //значения входного массива нужны
                for (int p = 0; p < n.veight.GetLength(0); p++)
                {
                    for (int q = 0; q < n.veight.GetLength(1); q++)
                    {
                        n.veight[p, q] += -GiperN * n.sigma * data[p, q];
                        n.sdvig += -GiperN * n.sigma * 1;
                    }
                }
            }
        }

        public List<List<Neiron>> testin()
        {
            return ArrayOfLayer;
        }
    }
}

```